The XMM-Newton ABC Guide for Hera

An Introduction To XMM-Newton Data Analysis Using Hera

NASA/GSFC XMM-Newton Guest Observer Facility

Lynne Valencic, Steve Snowden,
Brendan Perry, Michael Arida

With contributions by: Ilana Harrus, Stefan Immler, Rick Shafer, Randall Smith, Martin Still

Version 1.1

July 2011

Copies of this guide are available in `html`, `postscript` and `pdf` formats.

Contents

List of Tables

Table 1: List of Acronyms

ARF	Ancillary Region File
CAL	Calibration Access Layer
CCD	Charge Coupled Device
CCF	Current Calibration File
CIF	Calibration Index File
EPIC	European Photon Imaging Camera
FITS	Flexible Image Transport System
GO	Guest Observer
GOF	NASA/GSFC Guest Observer Facility
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HEASARC	High Energy Astrophysics Science Archive Research Center
HTML	Hyper Text Markup Language
OAL	ODF Access Layer
ODF	Observation Data File
OM	Optical Monitor
PDF	Portable Data Format
PP	Pipeline Processing System
PPS	Pipeline Processing
PV	Performance Validation
RGS	Reflection Grating Spectrometer
RMF	Redistribution Matrix File
SAS	Science Analysis System
SOC	Science Operations Center
SSC	Survey Science Centre
SV	Science Validation
XMM	X-ray Multi-Mirror Mission

Chapter 1

Introduction

The purpose of this Guide is to provide a simple walk-through of basic data extraction and analysis tasks for XMM-Newton data using SAS **as available through the Hera facility at HEASARC**. As with the original *XMM-Newton ABC Guide*, we have tried to balance providing enough information to give the user a useful introduction to a variety of analysis tasks with not providing too much information and thus overwhelming the user. This document is not intended to supercede the SAS Handbook, which is the highest authority for the use of SAS. Rather, this is meant as a general guide and introduction.

Chapters 7, 8, and 9 discuss the analysis of EPIC, RGS, and OM data respectively.

1.1 Acknowledgements

This guide would not have been possible without the help and comments from all people involved in the *X*MM-Newton project. In particular, we would like to thank Giuseppe Vacanti and Julian Osborne whose comments made this a more complete and accurate document.

IMH wishes to thank all the OM calibration team and in particular Antonio Talavera, Matteo Guainazzi and Bing Chen for their help in the preparation of this and other documents related to the OM.

SLS wishes to thank Dave Lumb, Richard Saxton, and Steve Sembay for their helpful insights into EPIC data analysis.

Chapter 2

Useful Information and References

2.1 Main Websites

- *X*MM-Newton SOC, fount of all *X*MM-Newton project information:

  `http://xmm.vilspa.esa.es/`

- NASA/GSFC GOF, source of US specific information and a mirror site for software and public data access:

  `http://xmm.gsfc.nasa.gov/`

- Survey Science Centre

  `http://xmmssc-www.star.le.ac.uk/`

2.2 *XMM-Newton* Help Desks

- The main project helpdesk is located at Vilspa and can be accessed through the WWW:

  `http://xmm.vilspa.esa.es/external/xmm_user_support/helpdesk.shtml`

  or via e-mail:

  `xmmhelp@xmm.vilspa.esa.es`

  The helpdesk also provides an archive of previously asked questions.

- The NASA/GSFC GOF offers an e-mail helpdesk for both general support and for US-specific issues:

  `xmmhelp@athena.gsfc.nasa.gov`

  Some questions addressed to the NASA/GSFC GOF may be redirected to the Vilspa helpdesk.

2.3 Mission Planning and Spacecraft Status

- Observation Log:

  `http://xmm.esac.esa.int/external/xmm_mission_plan/index.php`

  The scheduling information from this data base has been extracted and incorporated into a Browse data base at GSFC:

  `http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl`

- Long-Term Timeline:

  `http://xmm.vilspa.esa.es/external/xmm_sched/advance_plan.shtml`

2.4 Public Data Archives

- SOC Public Data Archive via the XSA:

  `http://xmm.vilspa.esa.es/external/xmm_data_acc/xsa/index.shtml`

- GSFC Archive Mirror Site via Browse:

  `http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl`

2.5 Software

- *XMM-Newton* Standard Analysis System (SAS):

  `http://xmm.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`

- HEASARC HEASoft Package:

  `http://heasarc.gsfc.nasa.gov/docs/corp/software.html`

- CXC CIAO Package:

  `http://asc.harvard.edu/ciao/`

2.6 Analysis, Documentation, and Helpful Hints

- On-Line SAS Handbook:

  `http://xmm.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`

  (click "Documentation", then click "its own documentation", then click "The SAS User's Guide"

- There is a "watchout" page for current SAS bugs at:

  `http://xmm.vilspa.esa.es/sas/7.1.0/watchout/`

- *XMM-Newton* Users Handbook:

  `http://xmm.vilspa.esa.es/external/xmm_user_support/documentation/uhb_frame.shtml`

- This Guide:

  `http://heasarc.gsfc.nasa.gov/docs/xmm/abc/`

- The MPE Analysis Guide:

  `http://www.mpe.mpg.de/xray/wave/xmm/cookbook/general/index.php`

- The Birmingham Analysis Guide (scripts etc. for EPIC extended source analysis):

  `http://www.sr.bham.ac.uk/xmm2/`

Chapter 3

A Few Words for the Veteran SAS Users

There is very little difference between processing data with Hera and SAS; that being the case, people who are already familiar with XMM-Newton data and SAS may find it useful to simply be aware of these differences, rather than go through this entire primer looking for discrepancies. For these users, the most relevant information pertains to the Hera GUI interface and placing data on the Hera servers (see §4). **Also, due to NASA network security restrictions, the pipeline task *epchain* cannot be used.** Further, the task *cifbuild* no longer needs to be run; the Hera server runs it every night, so the ccf.cif file is always up-to-date.

Users who generally prefer to run tasks from the command line may find that it is best to download their datasets from the HEASARC archive straight to their Hera account, repipeline the data there, and then analyze it using the Command Window instead of the GUI.

While tasks can also run on anonymous Hera, they take a **much** longer time, since the "files" in an anonymous user's account are really links; so when a user runs, say, *odfingest*, Hera has to replace the links with the files and unzip them before running the process.

Chapter 4

Hera

The Hera facility at HEASARC can run tasks from numerous environments, such as *CIAO*, *xselect*, and *ftools*, and includes the SAS environment, which is the software designed specifically for the reduction and analysis of XMM-Newton data. Hera lets users have access to this software remotely, without asking the users to download to the user's local machine such things as the calibration files or software. Data files stored on the user's local machine may be copied to a private space on the Hera server, or downloaded directly from the HEASARC data archive.

There are two flavors of Hera: anonymous and standard. These have identical graphical interfaces and produce identical outputs. Anonymous Hera is accessed though the HEASARC data archive by simply clicking on the "H" next to the dataset the user would like to examine. Files created through anonymous Hera will be deleted at the end of the session, but the user will be given the option to save the files to a new or existing Hera account. Standard Hera requires the user to install *fv*, available through the HEASARC, on his or her local machine (see §4.2). Standard Hera requires the user to log in to an account, and only processes data that is located on the Hera server. Files may be transferred there via anonymous Hera, HEASARC archive links, or by right-clicking the filename on the local directory list in the Hera GUI and dragging it to the desired location. Please note that only file at a time may be copied, and that due to security restrictions, whole directories cannot be copied (see §4.1).

There is no difference between the data analysis tasks in Hera's SAS and a locally-installed version of SAS – the only difference, as far as the user is concerned, is in the interface. Users who are familiar with SAS will need only to learn to navigate the Hera "wrapper"; all tasks, procedures, and techniques of data reduction are the same as if SAS were locally installed.

For the sake of simplicity, it will be assumed throughout this Guide that the user will only use SAS tasks (unless where explicitly stated). Thus, anonymous and standard Hera are synonomous with the locally-installed SAS GUI and command line.

4.1 Placing Data on the Hera Servers

Please be aware that network security regulations at NASA forbid the uploading of multiple files at once, or directories from local machines to Hera servers! Therefore, if you want to run tasks like the repipelining procedures (i.e., *emchain, emproc, epproc*), which require data from the ODF directory, you must download the ODF directory (and the PPS directory) to Hera disk space **directly from the HEASARC archive**. Users of standard Hera can click on the "Save to Hera" button in the Data Products Retrieval tab of the Browse Query Results. Users with Hera accounts will be prompted for their login information, while new users will be given the opportunity to make an account. For users of anonymous Hera, the data will be downloaded to a temporary directory for analysis. They will be given the option to save their data in an account at the session.

4.2 Setting Up and Running Standard Hera

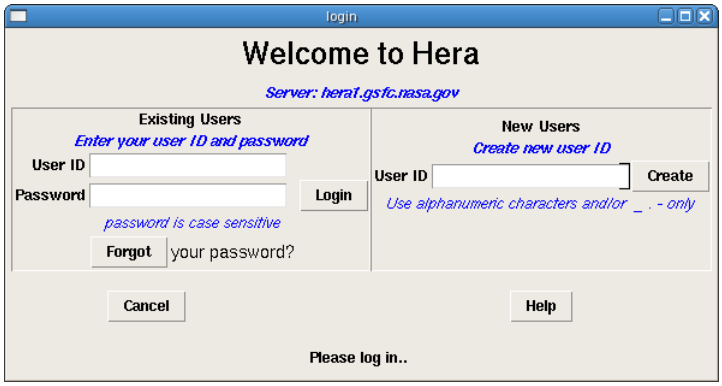
Hera currently runs on unix, Mac OS, and Windows. Unix users should install *fv* (version 4.4 or higher). Both Mac and Windows users will need to install a special Hera utility in addition to *fv*. Detailed information on downloading and installing the packages can be found at
`http://heasarc.gsfc.nasa.gov/docs/software/ftools/fv/fv_download.html`

The Hera GUI is called by either typing `fv -hera` on the command line or `fv &` on the command line and clicking on the "Connect to Hera..." option in the *fv* pop-up window, and logging in (see Figs. 4.1 and 4.2, respectively). If you do not yet have a Hera account, enter a username in the right-hand textbox and follow the instructions.

After logging in, the main Hera window will appear (see Figure 4.3). The main Hera window has five sections. The upper panel on the right side ("Remote Directory List") lists the directories and files on the NASA server in the user's account. Directly beneath it is a panel showing the directories and files on the user's local machine ("Local Directory List"). To the left are three smaller panels: the top lists any scripts the user may have ("Special Analysis Scripts"), the middle shows which packages and tasks are available ("Available Tools"), and the bottom lists which task is active, gives a short task description, and has the clickable "Help", "Run Tool...", and "Cmd Window" buttons. New accounts are created with a directory called **data** and two sample fits files, visible in the Remote Directory List window. Right-clicking on these files will bring up various options, including copying, deleting, and renaming them. Similarly, right-clicking on the data directory will bring up options including renaming the directory or creating a new one (under **data**).

Figure 4.1: The *fv* menu list.

Figure 4.2: The Hera login screen.

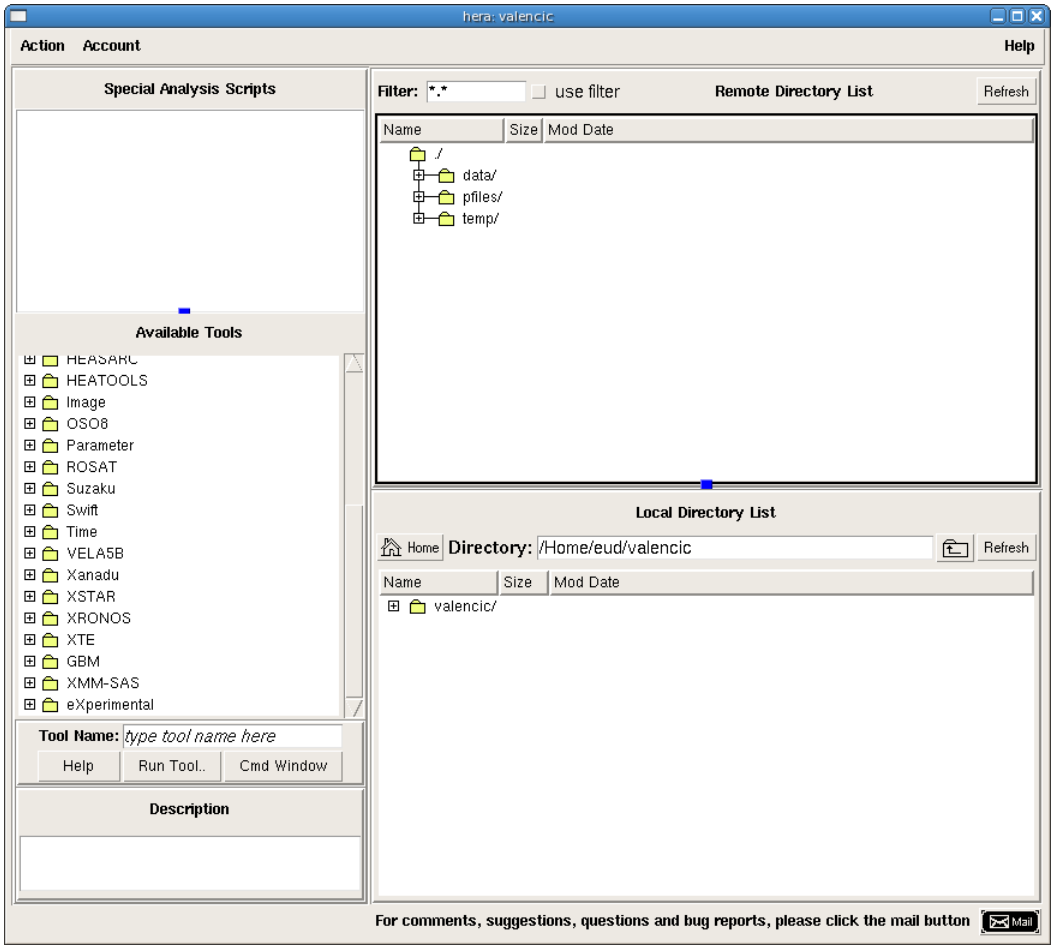


Mission-specific software is found by single-clicking on the mission name in the middle window. SAS tasks are listed under "XMM-SAS". Once a file has been selected in the Remote Directory List panel by single-clicking it, a task can be run on it by either single-clicking on the task name, then clicking "Run Tool...", or by double-

clicking the tool name. When a task is selected, its description will automatically appear in the "Description" box. Clicking on the "Help" button will pull up a website detailing the highlighted task, and clicking "Cmd Window" will cause a blank window with command line to pop up (see Fig. 4.4).

Running a task will cause a parameter window to appear, where the user can set the parameter values. When the parameters are set, clicking "Run" in the parameter window will run the task. If a Command Window is not already open, one will appear automatically, echoing the task name and all its parameter values. If Hera encounters problems while running the task, the warnings or error messages will be displayed there.

Figure 4.3: The main Hera GUI.



Command lines can often be quite long with a variety of parameters. To avoid considerable typing when creating command scripts, a feature of the standard Hera GUI interface can be of assistance: when invoking a task through the GUI, a copy of the full command appears in the Command Window; this can quickly be cut and pasted into a text file for future use.

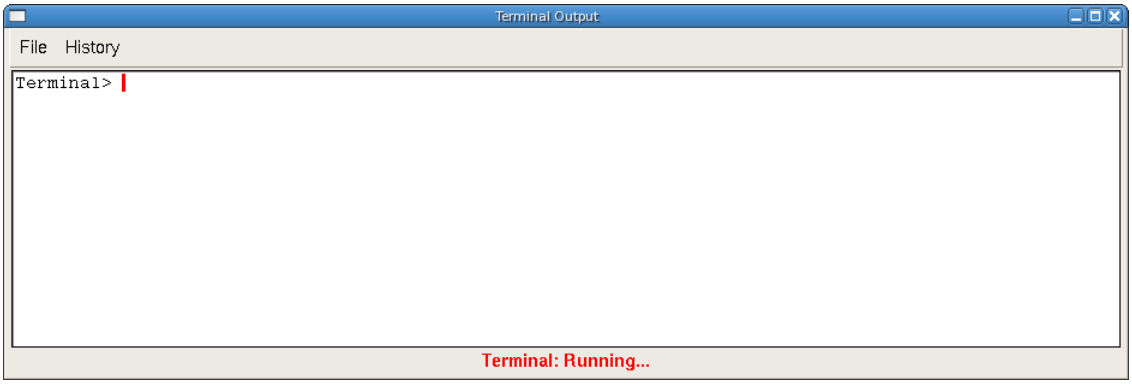
4.3 Command Line Hera Syntax and Logic

4.3.1 Table and Command Line Syntax

Veteran command-line SAS users, take note: there is some flexibility in command line Hera syntax, but less than what is in locally-installed SAS! First, all input tables in Hera are assumed to be event tables. Thus, there is no need to specify the EVENTS table when opening a file. The valid commands for opening a file in the Command Window are, therefore,

```
evselect table=filtered.fits
evselect table='filtered.fits'
```

Figure 4.4: The Command Window.




```
evselect table="filtered.fits"
```

The following syntaxes, while valid in the local installation of SAS, **are not valid** in Hera!

```
evselect table=filtered.fits:EVENTS
evselect table='filtered.fits:EVENTS'
evselect table="filtered.fits:EVENTS"
evselect table=filtered.fits%EVENTS
```

Further, the following are all valid task calls on the command line that result in identical operations:

```
evselect updateexposure=T
evselect updateexposure=yes
evselect updateexposure='yes'
evselect updateexposure="yes"
```

However,

```
evselect --updateexposure=T
evselect --updateexposure=yes
evselect --updateexposure='yes'
evselect --updateexposure="yes"
evselect -updateexposure=T
evselect -updateexposure=yes
evselect -updateexposure='yes'
evselect -updateexposure="yes"
```

are not correct syntax.

One format is not "more correct" than another, and the choice of which to use is left to user preference. In this guide we adopt the simplest format, and use no dashes and only single quotation marks only when required, such in denoting a list or filter, e.g.,

```
evselect filtertype=expression expression='(X,Y) in CIRCLE(26920,26400,500)'
```

4.3.2 Filtering Logic

Filtering event files requires some command of the SAS logical language which consists of familiar arithmetic and Boolean operators and functions. A full description can be found at the XMM-Newton SOC website, and will be explained in this Guide as they are used.

Chapter 5

Data

5.1 Useful Documentation

The documents most likely to be useful to the Hera user are those pertaining to SAS. There are a number of these which users of XMM-Newton data should be aware of. These documents include the *SSC Products Specification*, *Data Files Handbook*, *Reading Data Products CD's* (the most recent versions of these documents can be found in the SOC Document section under
`http://xmm.vilspa.esa.es/external/xmm_user_support/documentation/index.shtml`), and the *SAS Users Guide* (`http://xmm.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`).

Additional information concerning *X*MM-Newton data files can be found in the *I*nterface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM) (XMM-SOC-ICD-0004-SSD). This is an impressive tome which goes into great detail about the file nomenclature and structure. This document can be found in the documents area of the SOC web pages:
`http://xmm.vilspa.esa.es/cgi-bin/docs/DOC_list?Type=ICD`.

NOTE: For observation data sets going to US PIs, the GOF makes the data available on line after PGP encryption and after converting the file names to upper case. When the proprietary period for the observation expires the data are decrypted leaving the file names unchanged. A simple decryption script, minus the relevant keys of course, can be found at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/decrypt.pl`.
NOTE: Laura Brenneman wrote a script and accompanying help file that gives explicit directions on how to most quickly pull over all the files in a data set from the archive, as well as decrypting, and uncompressing the files in preparation for data analysis. This package can be found at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/prepare_xmm_data.tar.gz`. and contains the following files: `README`, `decrpyt.pl`, and `prepare_xmm_data.pl`.

5.2 The Data

One of the first steps that should be taken when examining your data is to check to see what you actually have. XMM-Newton observations can be broken into several exposures which are each assigned separate observation numbers. These separate exposures can be radically different in length and can also have the different instruments in different modes. For example, in one case the full observation was 60 ks with EPIC and RGS active but there was one delivered exposure which was ~ 3 ks and had only RGS active. (This can happen because the RGS can operate farther into regions of higher radiation than the EPIC detector. The additional observation time can be considered an additional exposure with only the RGS active.) Two files are useful and should be downloaded to the user's machine for this examination. First, the primary HTML page is `INDEX.HTM` which is included in the Pipeline Products. This page lists basic information for the observation plus the operational modes, filters, and exposure start and stop times for the individual instruments. It also has links to various summary pages, including those for the instruments. (In the case above, the EPIC summary page simply stated that "EPIC exposures processed by PPS None.") Specifically, **LOOK** at the `P*SUMMAR0000.HTM` files in the pipeline products (easily available through the links). Second, to quickly access images from the various instruments examine the PPSGRA Pipeline Products page (§ 5.3.3).

5.3 PI Data

Proprietary XMM-Newton data is available for download via your XSA account. Email instructions from the XMM-SOC at Vilspa are sent to the address on record with detailed directions on how to retrieve your data via the XSA.

XMM-Newton data is available for download at the HEASARC data archive. The data files can be considered to come in two groups in separate subdirectories when retrieved, the Observation Data Files (ODF) files and Pipeline Processing (PPS) files. The ODF data contain all of the observation-specific data necessary for reprocessing the observation. The PPS data contain, among other things, calibrated photon event files and source lists.

5.3.1 ODF Data

ODF data come with file names in the following format:

- `mmmm_iiiiiijjkk_aabeeeccfff.zzz`

  `mmmm` – revolution orbit number

  `iiiiii` – proposal number

  `jj` – target ID number in proposal

  `kk` – exposure number for target

  (**NOTE:** The ten-digit combination of `iiiiiijjkk` is the **observation number** and is used repetitively throughout the file nomenclature.)

  `aa` – detector (`M1` – MOS1, `M2` – MOS2, `PN` – PN, `OM` – OM, `R1` – RGS1, `R2` – RGS2, `SC` – spacecraft)

  `b` – flag for scheduled (`S`) or unscheduled (`U`) observations, or `X` for general purpose files)

  `eee` – exposure number

  `cc` – CCD number or OM window number

  `fff` – data identifier for the three detectors or spacecraft itself; see Table 5.1.

  `zzz` – format (`FIT` - FITS, `ASC` - ASCII)

5.3.2 Pipeline Product Data – Summary Files and Groupings

The Pipeline Processing (PP) produces quite a number of useful products which allow a first look at the data, but can overwhelm the user by their sheer numbers. The first place to look is the `INDEX.HTM` page which organizes the presentation of the data and provides links to other PP pages. The `INDEX.HTM` page also lists general observation information (target, date, time, etc.) and instrument modes.

The `INDEX.HTM` page provides links to various observation summary pages, which have names with the following nomenclature:

- `PPiiiiiijjkkAAX000SUMMAR0000.HTM`

  `iiiiiijjkk` – observation number

  `AA` – detector ID (`EP` - EPIC, `OM` - Optical Monitor, `RG` - RGS, `OB` - Observation)

PP data contain some immediately useful data products such as calibrated photon event lists, source lists, and images. While there are a large number of products which come in a single directory, they can be associated in up to 15 groupings; see Table 5.2. (The number of groups can vary depending on the number of operational instruments, e.g., if the OM is turned off there are no OM products.) Further information on each of these groupings and associated files, such as file contents, file types, and how they may be viewed, can be found in Table 5.3. Each group has an associated HTML file which organizes access to the files and provides a limited description of them. The names of the HTML files are of the following form:

- `PPiiiiiijjkkAAAAAA000_0.HTM`

  `iiiiiijjkk` – observation number

  `AAAAAA` – group identifier (see Table 5.2)

5.3.3 Pipeline Product Data – Data Files

The data file names are of the form (see Table 41 in the *XMM Data Files Handbook*):

- `Piiiiiijjkkaabll lCCCCCCnmmm.zzz`

  `iiiiiijjkk` – observation number

  `aa` – detector, `M1` - MOS1, `M2` - MOS2, `PN` - PN, `CA` - for files from the CRSCOR group, `R1` - RGS1, `R2` - RGS2, `OM` - OM.

  `b` – flag for scheduled (`S`) or unscheduled (`U`) observations, or `X` for files from the CRSCOR group (and any product that is not due to a single exposure)

  `lll` – exposure number

  `CCCCCC` – file identification for data from each detector; see Table 5.3

  `n` – For EPIC data, this is the exposure map band number; for RGS data, this is the spectral order number; for the OM, this is the OM window within the exposure.

  `mmm` – source number in hexidecimal

  `zzz` – file type (e.g., PDF, PNG, FTZ, HTM)

  - `ASC` - ASCII file, use a web browser, or the "more" command
  - `ASZ` - gzipped ASCII file
  - `FTZ` - gzipped FITS format, use *ds9, Ximage, Xselect, fv*
  - `HTM` - HTML file, use Firefox or other web browser
  - `PDF` - Portable Data Format, use *Acrobat Reader*
  - `PNG` - Portable Networks Graphics file, use a web browser
  - `TAR` - TAR file

Table 5.1: ODF data file identifiers.

Data ID	Contents
	EPIC files
IME	Event list for individual CCDs, imaging mode
RIE	Event list for individual CCDs, reduced imaging mode
CTE	Event list for individual CCDs, compressed timing mode
TIE	Event list for individual CCDs, timing mode
BUE	Event list for individual CCDs, burst mode
AUX	Auxiliary file
CCX	Counting cycle report (auxiliary file)
HBH	HBR buffer size, non-periodic housekeeping
HCH	HBR configuration, non-periodic housekeeping
HTH	HBR threshold values, non-periodic housekeeping
PEH	Periodic housekeeping
PTH	Bright pixel table, non-periodic housekeeping
DLI	Discarded lines data
PAH	Additional periodic housekeeping
PMH	Main periodic housekeeping
	RGS files
AUX	on-board processing statistics
SPE	raw event list for one CCD
DII	diagnostic images
D1H	CCD readout settings
D2H	CCD readout settings
PFH	housekeeping data
ODX	pixel offset data
	XMM files
ATS	spacecraft attitude history
	OM files
IMI	imaging file
THX	tracking history file
WDX	window data auxiliary file
NPH	non-periodic housekeeping file
PEH	periodic housekeeping file
PAX	field acquisition data
RFX	priority reference frame data
PFX	priority fast mode data
FAE	event list (if fast mode was used)

Table 5.2: Pipeline Processing groupings.

Group ID	Contents
	PP files
PPSDAT	Contains the Calibration Index File (CIF) used in the pipeline processing (*CALIND*), PPS information, and the attitude history time series (*ATTTSR*) in gzipped FITS or ASCII format.
PPSGRA	Contains the OM tracking history plots, PPS, EPIC, OM, RGS observation, and PPS run summaries. **NOTE: CHECK THESE OUT**
PPSMSG	ASCII file containing pipeline processing report
	EPIC files
CRSCOR	Contains PDF files of POSS II finding charts, HTML files of cross correlations with the SIMBAD data base, FITS tables for the detected sources
EANCIL	Contains the exposure maps in a variety of energy bands and the source-detection sensitivity maps for the EPIC instruments. The sensitivities are in units of counts s^{-1} corrected for vignetting and corresponding to a likelihood specified in the FITS header. The files are gzipped with a .FTZ extension.
EEVLIS	Contains calibrated photon event files for the EPIC detectors. If the files are sufficiently large they may be separated into two tar files. The files are gzipped fits files with a .FTZ extension.
ESKYIM	This group contains the event images in a variety of energy bands. The fits files are gzipped with a .FTZ extension, the full images also come as PNG images.
ESRLIS	Contains EPIC observation source lists. There is an HTML page of the merged source list and gzipped fits tables of source lists from the different instruments and source detection tasks.
	OM files
OIMAGE	Contains OM sky images in gzipped FITS format.
OMSLIS	Contains OM observation source lists in gzipped FITS format.
OMSRTS	Contains OM star tracking time series in gzipped FITS format.
	RGS files
REVLIS	Contains the RGS source and event lists in gzipped FITS format
REXPIM	Contains the RGS exposure maps in gzipped FITS format
RIMAGE	Contains the RGS images (both energy dispersion and cross dispersion) in gzipped FITS and PNG formats
RSPECT	Contains the RGS source and background spectra in gzipped FITS and PDF formats

Table 5.3: Pipeline Processing data files

Group ID	File ID	Contents	File Type	View With
		EPIC files		
CRSCOR	FCHART	Finding chart	PDF	*Acrobat Reader*
	ROSIMG	ROSAT image of region	PDF	*Acrobat Reader*
	SNNNNN[1]	Source cross-correlation results	Zipped FITS	*fv*
	DNNNNN[1]	Catalog descriptions	PDF	*Acrobat Reader*
	FNNNNN[1]	FOV cross-correlation result	Zipped FITS	*fv*
ESKYIM	IMAGE_8	Sky image 0.2 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_1	Sky image 0.2 - 0.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_2	Sky image 0.5 - 2.0 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_3	Sky image 2.0 - 4.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_4	Sky image 4.5 - 7.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_5	Sky image 7.5 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
EANCIL	EXPMAP_8	Exposure map 0.2 - 12.0 keV	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
	EXPMAP_1	Exposure map 0.2 - 0.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_2	Exposure map 0.5 - 2.0 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_3	Exposure map 2.0 - 4.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_4	Exposure map 4.5 - 7.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_5	Exposure map 7.5 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
EEVLIS[2]	MIEVLI	MOS imaging mode event list	Zipped FITS	*xmmselect, fv, Xselect*
	PIEVLI	PN imaging mode event list	Zipped FITS	*xmmselect, fv, Xselect*
	TIEVLI	PN, MOS timing mode event list	Zipped FITS	*xmmselect, fv, Xselect*
ESRLIS	EBLSLI	Box-local detect source list	Zipped FITS	*fv*
	EBMSLI	Box-map detect source list	Zipped FITS	*fv*
	EMSRLI	Max-like detect source list	Zipped FITS	*fv*
	OBSMLI	Summary source list	Zipped FITS, HTML	*fv*, web browser
		RGS files		
REVLIS	SRCLI_	RGS Source Lists	Zipped FITS	*fv*
	EVENLI	RGS Event lists	Zipped FITS	*xmmselect, fv*
REXPIM	EXPMAP	RGS Exposure Maps	Zipped FITS	*ds9, Ximage, fv*
RSPECT	SRSPEC1	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	SRSPEC2	2nd Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC1	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC2	2nd Order Source Spectra	Zipped FITS	*Xspec, fv*
	SRSPEC	Spectra Plots	PDF format	*Acrobat Reader*
RIMAGE	ORDIMG	Images, disp. vs. X-disp	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
	IMAGE_	Images, disp. vs. PI	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
		OM files		
OANCIL	SWSREG	OM Region File	ASCII text	text editor
OIMAGE	SIMAGE	OM Sky Image	Gzipped FITS	*ds9, Ximage, fv*
OMSLIS	SWSRLI	OM Source Lists	Zipped FITS	*fv*
OMSRTS	TSTRTS	Tracking Star Time Series	Zipped FITS	*fv*

[1] NNNNN - Alphanumeric ID

[2] Files for only those modes which were active will be included

Chapter 6

Preparing the Data for Processing

Throughout this Primer, data from the HEASARC archive are used to illustrate how to run tasks for each instrument on XMM-Newton; new users are encouraged to use these sample data, though in should be noted that any data from the relevant instrument will do. Information about the example datasets are in Table 6.1.

Regardless of which instrument you are interested in, one task is necessary to prepare the data for processing, *odfingest*. To demonstrate *odfingest*, and a host of other tasks in §7, we will use the Lockman Hole data with ObsID 0147511601, though any dataset will work equally well.

To begin, either run anonymous Hera by clicking on the "H" next to the dataset in the archive (see §4), or download the dataset from the HEASARC archive and initialize standard Hera (see §4.1 and §4.2, respectively). Select all the zipped files in the ODF directory and unzip them; right-clicking on the selection will bring up the option. (This takes a long time.)

The task *odfingest* extends the Observation Data File (ODF) summary file with data extracted from the instrument housekeeping data files and the calibration database. It is required for reprocessing the ODF data with the pipeline tasks, as well as for many other tasks, and it is only necessary to run it once on any dataset. If for some reason *odfingest* must be rerun, you must first delete the earlier file it produced. This file largely follows the naming convention described in §5.3.3, but has `SUM.SAS` appended to it. Hera will automatically set the environment parameter `SAS_ODF` to the `SUM.SAS` file in the active dataset's ODF directory, if one exists. If there is no `SUM.SAS` file, the `SAS_ODF` variable maintains its default value (the dataset's ODF directory).

To run *odfingest* from the GUI:

1) Highlight the ODF directory on the Hera server that you are interested in by single-clicking on it.

2) In the Available Tools panel, under "XMM-SAS", call *odfingest*.

3) In the parameter pop-up window, click "Run".

To run *odfingest* from the Command Window, type:

```
odfingest
```

In the Command Window, the command and parameters will be echoed, and dialog from the task will be shown. The warnings can be ignored. The prompt will return when it is complete.

At this point, the data is ready to be repipelined and analyzed. Data from the EPIC camera is discussed in §7, the RGS is in §8, and the OM is in §9.

Table 6.1: Example datasets used in this Guide.

Instrument	Chapter	ObsID	Object
EPIC	7	0147511601	Lockman Hole
EPIC (Timing mode)	7.6	0122700101	G21.5-09
RGS	8	0134520301	AB Dor
OM (Image mode)	9.3	0123700101	Lockman Hole
OM (Fast mode)	9.4	0411081601	Mkn 421
OM (Grism mode)	9.5	0125320801	BPM 16274

Chapter 7

An EPIC Data Processing and Analysis Primer

While a variety of analysis packages can be used for the following steps, the SAS was designed for the basic reduction and analysis of XMM-Newton data (extraction of spatial, spectral, and temporal data); therefore, it will be used here for demonstration purposes.

NOTE: For PN observations with very bright sources, out-of-time events can provide a serious contamination of the image. Out-of-time events occur because the read-out period for the CCDs can be up to $\sim 6.3\%$ of the frame time. Since events that occur during the read-out period can't be distinguished from others events, they are included in the event files but have invalid locations. For observations with bright sources, this can cause bright stripes in the image along the CCD read-out direction. For a more detailed description of this issue, check: `http://wave.xray.mpe.mpg.de/xmm/cookbook/EPIC_PN/ootevents.html`

It is **strongly** recommended that you keep all reprocessed data in its own directory! Hera places output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipleline Processed data in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, and the analysis is taking place in the PROC directory.

At this point, it is assumed that you have downloaded the data from the HEASARC archive onto a Hera server, standard or anonymous Hera is running (see §4.2), and you have prepared the data for processing (see §6). Throughout this chapter, as in §6, we will use the Lockman Hole dataset with ObsID 0147511601, though any dataset will suffice.

7.1 A Quick Look at What You Have

The PP source lists for each detector (`*EVLI*`) are provided in zipped FITS format. These can be viewed through Hera with *fv* by double-clicking on the event file.

7.2 Rerunning the Pipeline

If a dataset is more than a year old, it was probably processed with older versions of CCF and SAS prior to archiving, so the pipeline should be rerun to generate event files with the latest calibrations. The MOS has two pipeline tasks, *emchain* and *emproc*, while the PN has one, *epproc* for the PN. The two MOS tasks produce the same output, so which one to use is entirely a matter of the user's personal preference.

First, create the directory for processed data, PROC, by right-clicking on the ObsID directory and choosing the option "Create new directory". Highlight PROC by single-clicking on it. Please note that both the Hera GUI and Command Window rely on this to know where to place the output files.

To rerun the pipeline from the GUI:

1) In the Available Tools panel, under "XMM-SAS", call *emproc* or *emchain* to repipeline the MOS data, and *epproc* to repipeline the PN data. Note that for *emchain*, the default is to process both the MOS1 and MOS2 cameras; users who want to analyze only one camera will need to change the "Instrument to analyze" parameter. In the task pop-up window, click "Run".

2) Rename the output something easy to type, for example, `mos1.fits`, by right-clicking on the repipelined event file. For *emproc* and *epproc*, the files with have the form `*EMOS1*ImagingEvts.ds, *EMOS2*ImagingEvts.ds`, and `*EPN*ImagingEvts.ds`. For *emchain*, the output files follow the naming system as described in §5.3.2.

To rerun the pipeline from the Command Window, type:

`emchain` (or `emproc`)

`epproc`.

We will assume that the newly pipelined event files are named `mos1.fits`, `mos2.fits`, and `pn.fits`.

7.3 Examine and Filter the Data

Since the event files are current, we can proceed with some simple analysis demonstrations. The following sections describe the use of Hera tasks using the both the command line and GUI interfaces, except in cases where one of the methods is particularly easy. People new to Hera (or SAS) will likely prefer the GUI, at least at first; however, as they become more familiar with the software and the keywords, they will probably migrate to the command line, which is faster. Assuming that the parameter values for any given task are the same, it does not matter if a task is invoked on the command line or in the GUI; the output files will be identical. The *heraXmmselect* GUI, in the "experimental" folder of "XMM-SAS", provides a very simple method for producing and displaying images, spectra, and light curves, and is the recommended method for extracting data unless large numbers of sources are being analyzed.

An event list from the MOS1 can be loaded by highlighting the event file `mos1.fits` in the PROC directory, then double-clicking on *heraXmmselect*. A Table Selection window will pop up to confirm your selection; clicking "Go" will start *heraXmmselect*. If using the Command Window, be sure it knows which directory you are working in by highlighting the directory in the GUI.

7.3.1 Create and Display an Image

To create an image in sky coordinates by using the *heraXmmselect* GUI:

1) Check the square boxes to the left of the "X" and "Y" entries.

2) Click on the "Image" button near the bottom of the page. This brings up the *evselect* GUI (see Figure 7.1).

3) In the `imageset` box (under the `Image` tab), enter the name of the output file, in this case, `image.fits`.

4) Click on the "Run" button on the lower left corner of the *evselect* GUI.

Different binnings and other selections can be invoked by accessing the "Image" tab at the top of the GUI. The default settings are reasonable, however, for a basic image. The resultant image is written to the file `image.fits`, and is automatically displayed with *POW*. It can also be downloaded to your local machine and viewed with *ds9*; see Figure 7.2.

To create an image in sky coordinates by using the Command Window, type:

```
evselect table=mos1.fits withimageset=yes imageset=image.fits
   xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
```

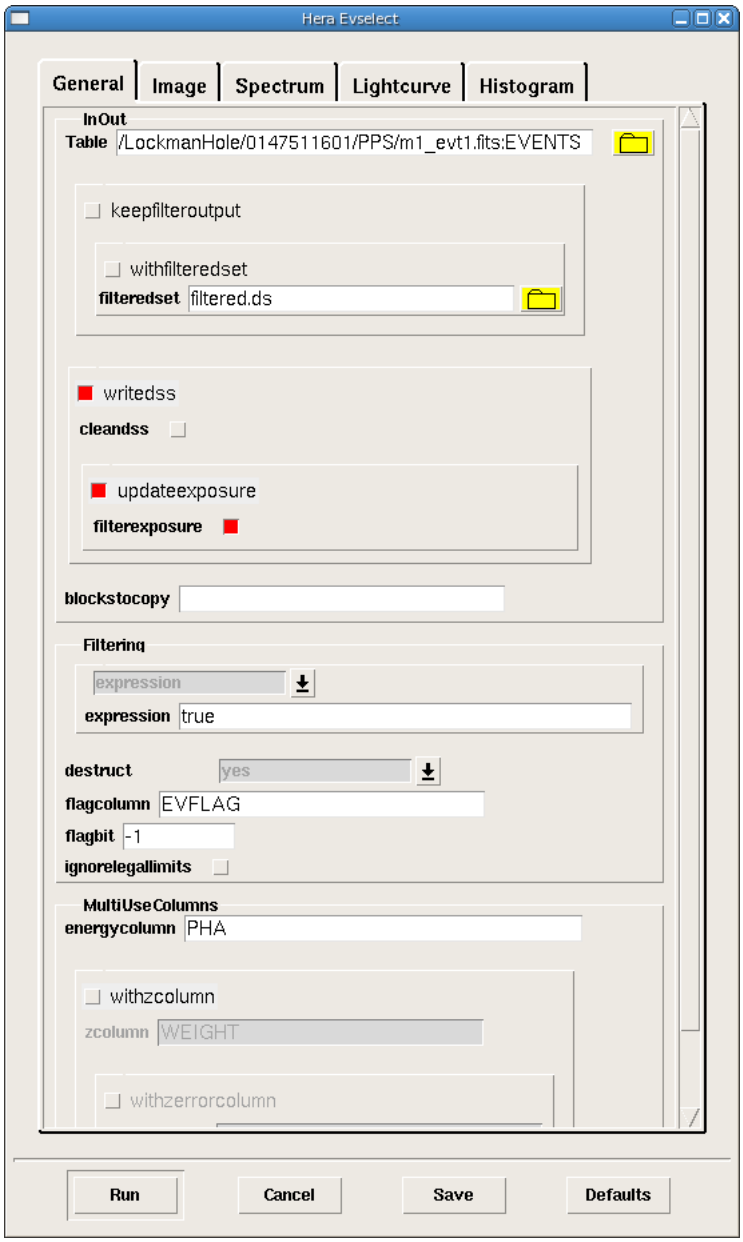
where

`table` – input event table
`withimageset` – make an image
`imageset` – name of output image
`xcolumn` – event column for X axis
`ycolumn` – event column for Y axis
`imagebinning` – form of binning, force entire image into a given size or bin by a specified number of

pixels
`ximagesize` – output image pixels in X
`yimagesize` – output image pixels in Y

The resultant image is written to the file `image.fits`. It can be viewed with *POW*, or downloaded to your local machine and viewed with *ds9*; see Figure 7.2.

Figure 7.1: The *evselect* GUI.

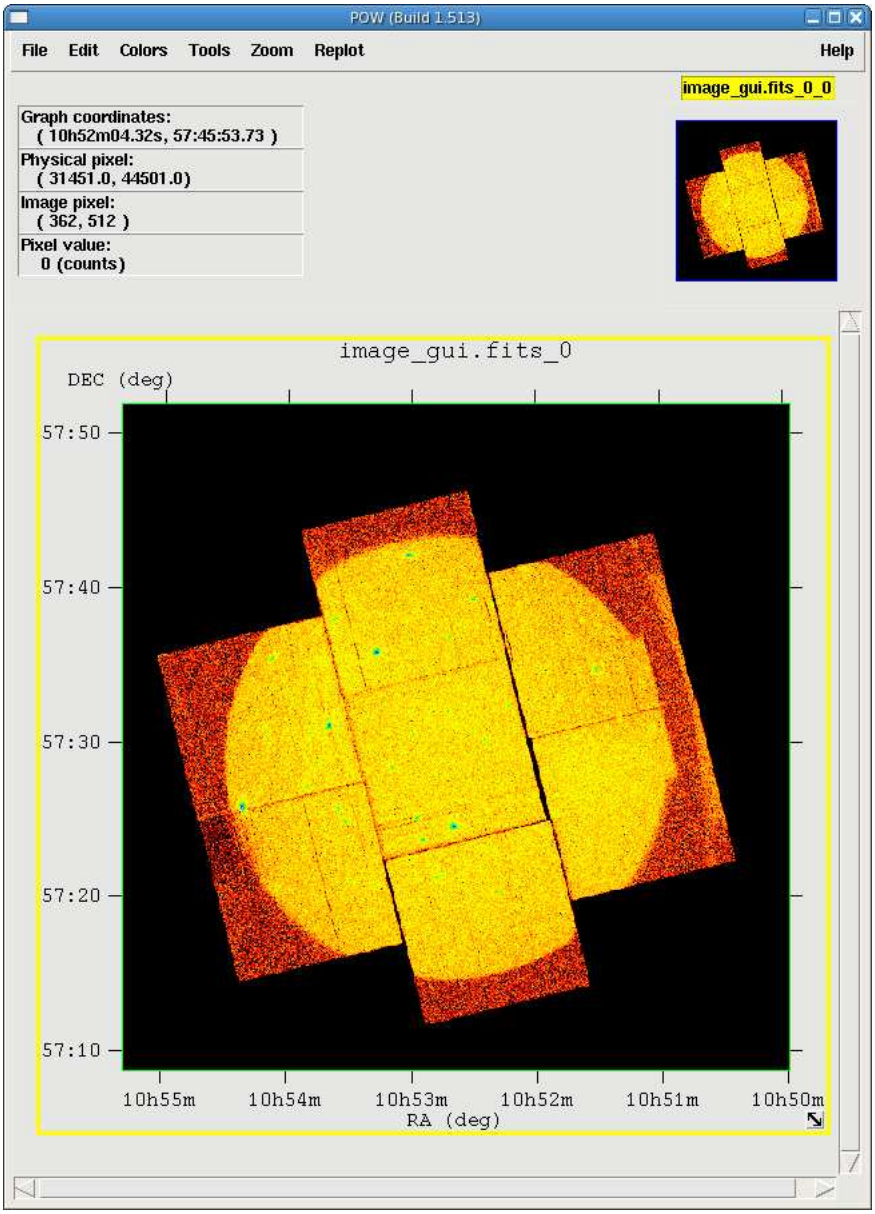


7.3.2 Create and Display a Light Curve

To create a light curve by using the *heraXmmselect* GUI:

1) Check the diamond to the left of the "Time" entry.

Figure 7.2: The MOS1 image, displayed in *fv*.



2) Click on the "OGIP Rate Curve" button near the bottom of the page. This brings up the *evselect* GUI (see Figure 7.1).

3) Click on the "Lightcurve" tab and change the "timebinsize" to a reasonable amount, e.g. 10 or 100 s. Change the name of the output file in the "rateset" box to, say, `mos1_ltcrv.fits`.

4) Click on the "Run" button at the lower left corner of the *evselect* GUI.

The resultant light curve is written to the file `mos1_ltcrv.fits` and is automatically displayed in *POW*; see Fig. 7.3.

To create a light curve by using the Command Window, type:

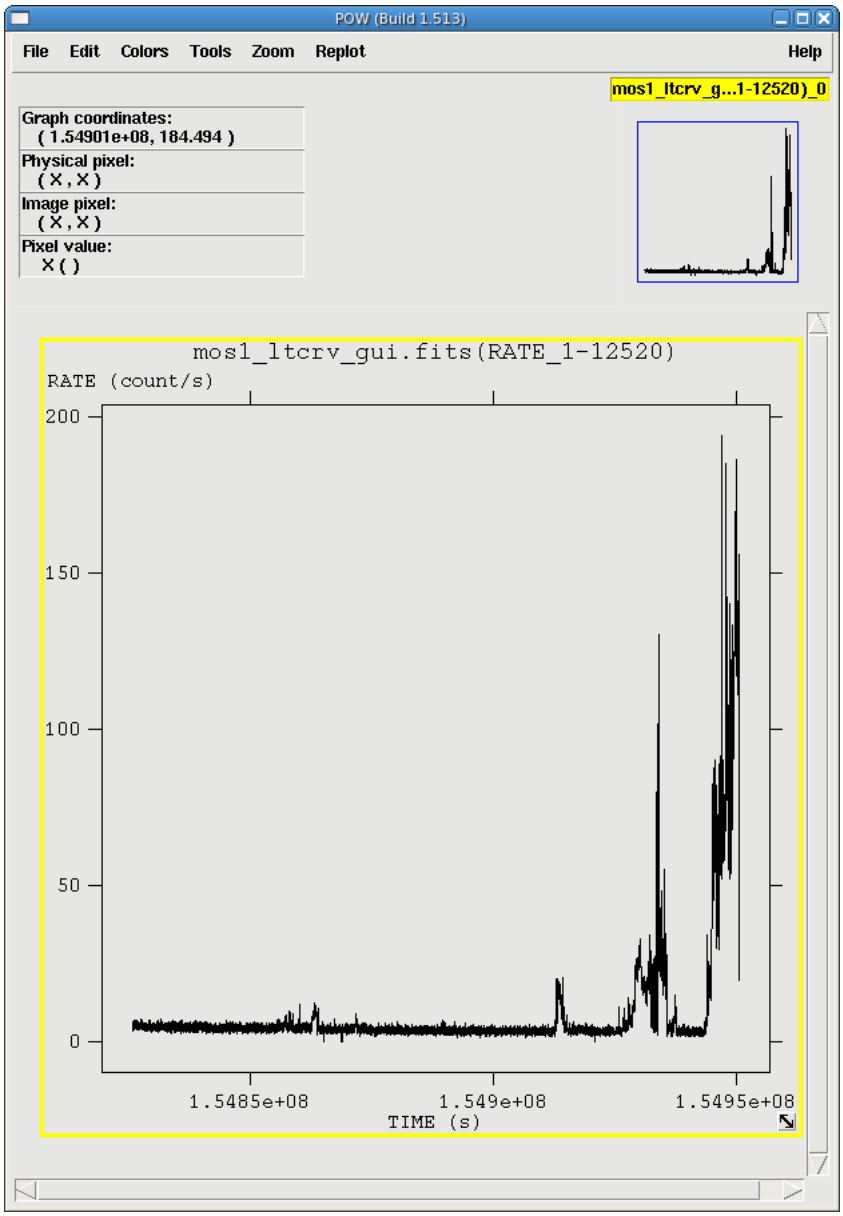
```
evselect table=mos1.fits withrateset=yes rateset=mos1_ltcrv.fits
   maketimecolumn=yes timecolumn=TIME timebinsize=100 makeratecolumn=yes
```

where

`table` – input event table
`withrateset` – make a light curve
`rateset` – name of output light curve file
`maketimecolumn` – control to create a time column
`timecolumn` – time column label
`timebinsize` – time binning (seconds)
`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

The output file `mos1_ltcrv.fits` can be viewed with *POW*; see Fig. 7.3.

Figure 7.3: The light curve.



7.3.3 Applying Standard Filters the Data

Whether using the GUI or the command line, the needed filtering "expressions" for the MOS and PN are, respectively:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
```

and

```
(PATTERN <= 12)&&(PI in [200:15000])&&#XMMEA_EP
```

If the PN data is timed, then the `PATTERN` parameter should be set to 4:

```
(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP.
```

The first two expressions will select good events with `PATTERN` in the 0 to 12 range, and the last will select events with `PATTERN` between 0 and 4. The `PATTERN` value is similar the `GRADE` selection for ASCA data, and is related to the number and pattern of the CCD pixels triggered for a given event.The `PATTERN` assignments are: single pixel events: `PATTERN == 0`, double pixel events: `PATTERN in [1:4]`, triple and quadruple events: `PATTERN in [5:12]`.

The second keyword in the expressions, `PI`, selects the preferred pulse height of the event; for the MOS, this should be between 200 and 12000 eV. For the PN, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower PI channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the `#XMMEA_EM` (`#XMMEA_EP` for the PN) filter provides a canned screening set of FLAG values for the event. (The FLAG value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting `FLAG == 0` in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done.

It is a good idea to keep the output filtered event files and use them in your analyses, as opposed to re-filtering the original file with every task. This will save much time and computer memory. As an example, the pipelined Lockman Hole data for the MOS1 is 44Mb; the quality-filtered list is 29Mb, and when the temporal filter is applied, the final "clean" event list is only 7Mb!

To filter the data using the *heraXmmselect* GUI:

1) Enter the filtering criteria in the "Selection Expression" area at the top of the *heraXmmselect* GUI:

   ```
   (PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
   ```

2) Click on the "Filtered Table" box at the lower left of the *heraXmmselect* GUI.

3) In the General tab, change the `filteredset` parameter, the output file name, to something useful, e.g., `mos1_filt.fits`

4) Click "Run".

To filter the data using *evselect* in the Command Window, type:

```
evselect table=mos1.fits withfilteredset=yes
   expression='(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM'
   filteredset=mos1_filt.fits filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where

`table` – input event table
`filtertype` – method of filtering
`expression` – filtering expression
`withfilteredset` – create a filtered set
`filteredset` – output file name
`keepfilteroutput` – save the filtered output
`updateexposure` – for use with temporal filtering
`filterexposure` – for use with temporal filtering

7.3.4 Applying Time Filters the Data

Sometimes, it is necessary to filter on time, in addition to those mentioned above. This is because of soft proton background flaring, which can have count rates of 100 counts/sec or higher.

It should be noted that the amount of flaring that needs to be removed depends in part on the object observed; a faint, extended object will be more affected than a very bright X-ray source.

There are two ways to filter temporally: with an explicit reference to the `TIME` or `RATE` parameters in the filtering expression, or by creating a secondary Good Time Interval (GTI) file with the task *tabgtigen*. Both procedures are described below. For the example data, we will filter by time, though you can just as easily filter by rate.

To explicitly define the `TIME` or `RATE` parameters in the GUI or Command Window:

1) Make a light curve and display it in *fv* (see §7.3.2). In the *fv* window, you can zoom in on interesting sections by making a selection box by left-clicking over the graph, beginning with the upper left hand corner of the region.

2) Mouse over the plot to see the rates and times, and note the times for the "good" regions.

The syntax for the time selection is `(TIME < 1.54912e8)`. A more complicated expression which would remove some small flares within an otherwise good interval (e.g., the soft proton flares observed in the light curve plot of Figure 7.3) could be: `(TIME < 1.54912e8)&&!(TIME IN [1.54854e8:1.54872e8])`. The syntax `&&(TIME < 1.54912e8)` includes only events with times less than 1.54912e8, and the "!" symbol stands for the logical "not". So, use `&&!(TIME in [1.54854e8:1.54872e8])` to exclude events from the interval 1.54854e8 to 1.54856e8. If combined with the standard filtering expression (see §7.3.2), the full filtering expression would then be:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
   &&(TIME < 1.54912e8)&&!(TIME in [1.54854e8:1.54872e8])
```

This expression can then be used in either the *heraXmmselect* GUI or Command Window to filter the event file, as in §7.3.3. If using the GUI, be certain that the `updateexposure` and `filterexposure` boxes are checked. Similarly, if using the Command Window, make sure to set those parameters to "yes", as in the example above. As ever, give the new file a useful name; here, we will use `mos1_filt_time.fits`.

To make a secondary GTI file and apply it in the GUI:

1) Call *tabgtigen* from the Available Tools panel and enter the light curve, `mos1_ltcrv.fits` as the input data set. (There is no need to quit *heraXmmselect*.)

2) Edit the name of table to hold resulting GTI list; here, we will use `gtiset.fits:STDGTI`.

3) In the text box directly below ("Boolean expression controlling the GTI creation"), enter the filtering expression . In this case, it is `(TIME < 1.54912e8) &&!  (TIME in [1.54854e8:1.54872e8])`

4) Click "Run" in the lower left corner.

5) In *heraXmmselect*, load the filtered event file by going to File → New Table and entering `mos1_filt.fits`.

6) In the "Selection Expression" box, type `GTI(gtiset.fits,TIME)`.

7) Click on the "Filtered Table" box at the lower left of the *heraXmmselect* GUI.

8) Change the `filteredset` parameter, the output file name, to something useful; here, we will use `mos1_filt_time.fits`.

9) Click "Run".

To make a secondary GTI file and apply it in the Command Window, type:

```
tabgtigen table=mos1_ltcrv.fits gtiset=gtiset.fits timecolumn=TIME
   expression='(TIME < 1.54912e8)&&!(TIME in [1.54854e8:1.54872e8])'
```

where

`table` – input count rate table and extension (see §7.3.2)
`expression` – filtering expression
`gtiset` – output file name for selected GTI intervals
`timecolumn` – time column

And then type:

```
evselect table=mos1_filt.fits withfilteredset=yes
   expression='GTI(gtiset.fits,TIME)' filteredset=mos1_filt_time.fits
   filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where

`table` – input count rate table and extension (see §7.3.2)
`expression` – filtering expression
`withfilteredset` – create a filterered set
`filteredset` – output file name
`filtertype` – method of filtering
`keepfilteroutput` – save the filtered set
`updateexposure` – update exposure information in event list and in spectrum files
`filterexposure` – filter exposure extensions of event list with same time filters as for corresponding CCD

7.4 Extract and Fit the Spectrum

7.4.1 Extract the Source Spectrum

Throughout the following, some parameters are instrument-dependent! **The parameter `spectralbinsize` should be set to 15 if you are using the MOS; if you are using the PN, it should be 5. The parameter `specchannelmax` should be set to 11999 for the MOS, or 20479 for the PN.** Also, remember that the most stringent filter, `FLAG==0`, must be applied to get a high-quality spectrum.

To extract the source spectrum using the GUI:

1) Click on *heraXmmselect* and open the filtered file, `mos1_filt_time.fits`.
2) Make an image (see §7.3.1). It will be displayed automatically in *POW*.
3) Click on the object whose spectrum you wish to extract. This will produce a circle (extraction region), centered on the object. The circle's radius can be changed by clicking on it.
4) Adjust the size and position of the circle until you are satisfied with the extraction region.
5) Click on "2D Region" in the *heraXmmselect* GUI. This transfers the region information into the "Selection Expression" text area, for example, `((X,Y) IN CIRCLE(25705.5,18820.5,400))`; the `CIRCLE` parameters are the X, Y coordinates of the center, and the radius is in units of 0.05 arcsec, so our extraction circle has a radius of 20". In the "Selection Expression" box, enter `&& (FLAG==0)`.
6) Click the diamond next the `PI` column on the *heraXmmselect* GUI.
7) Click on "OGIP Spectrum".
8) In the "General" page, check `keepfilteroutput` and `withfilteredset`. In the `filteredset` box, enter the name of the event file output, in this case, `mos1_filt_time_source.fits`.
9) Select the "Spectrum" tab of the *evslect* GUI to set the file name and binning parameters for the spectrum. Confirm that `withspectrumset` is checked. Set `spectrumset` to the desired output name, in this case, `mos1_source_pi.fits`. Confirm that `spectralbinsize` is set to the correct value: 15 for the MOS, 5 for the PN. Confirm that `withspecranges` is checked. Set `specchannelmin` to 0. Set `specchannelmax` to 11999 for the MOS, or 20479 for the PN.

10) Click "Run".

A plot of the counts per channel will be displayed automatically in *fv*. (The flux per energy cannot be calculated until the response matrix and ancillary files are made, see §7.4.5.)

To extract the source spectrum using the Command Window:

1) Make an image of the filtered file, `mos1_filt_time.fits`, as described in §7.3.1). It will be displayed automatically in *POW*.
2) Click on the object whose spectrum you wish to extract. This will produce a circle (extraction region), centered on the object. The circle's radius can be changed by clicking on it.
3) Adjust the size and position of the circle until you are satisfied with the extraction region.
4) Double-click on the chosen region. This will bring up a window showing the (X, Y) center coordinates and radius of the circle. For example, lets say the center is at (25705.5,18820.5) and the radius is 400.
5) In the Command Window, type the following. Remember that for PN data, for PN data, `spectralbinsize` must be set to 5.

```
evselect table='mos1_filt_time.fits' energycolumn='PI' withfilteredset=yes
   filteredset='mos1_filt_time_source.fits' keepfilteroutput=yes filtertype='expression'
   expression='((X,Y) in CIRCLE(25705.5,18820.5,400))&&(FLAG==0)'
   withspectrumset=yes spectrumset='mos1_source_pi.fits' spectralbinsize=15
   withspecranges=yes specchannelmin=0 specchannelmax=11999
```

where

`table` – the event file
`energycolumn` – energy column
`withfilteredset` – make a filtered event file
`keepfilteroutput` – keep the filtered file
`filteredset` – name of output file
`filtertype` – type of filter
`expression` – expression to filter by
`withspectrumset` – make a spectrum
`spectrumset` – name of output spectrum
`spectralbinsize` – size of bin, in eV
`withspecranges` – covering a certain spectral range
`specchannelmin` – minimum of spectral range
`specchannelmax` – maximum of spectral range

The spectrum, in counts per channel, can be viewed with *fv*.

Next, the task *backscale* finds the area of the source region, taking into account any bad pixels or chip gaps, and writes the area into the BACKSCAL keyword of the spectrum table.

To find the spectrum's source area using the GUI:

1) Call *backscale* in the Available Tools panel and enter the spectrum, `mos1_source_pi.fits` as the input data set.
2) Directly beneath, enter the file containing the bad pixel extensions, i.e., the event file from which it was extracted, `mos1_filt_time.fits`. Confirm that the option to correct for bad pixels in the source box is set to "yes".
3) Click "Run".

To find the spectrum's source area using the Command Window, type:

```
backscale spectrumset=mos1_source_pi.fits
withbadpixcorr=yes badpixlocation=mos1_filt_time.fits
```

7.4.2 Extract the Background Spectrum

To extract the background spectrum using the GUI:

1) Follow steps 1 - 3 of §7.4.1 (GUI instructions).

2) Make an annulus around the object whose background spectrum you wish to extract. This can be done using two circles, each defining the inner and outer edges of the annulus. The region in the inner circle can be excluded by selecting it, and then clicking the "-" sign in the Edit Region window, and "Apply".

3) Adjust the sizes and positions of the circles until you are satisfied with the extraction region. In this case, we will center the annulus on the spectrum extraction region from §7.4.1 and let the background region extend from 20" to 1' around the source.

4) Click on "2D Region" in the *heraXmmselect* GUI.

5) Confirm that the correct values were transferred to the "Selection Expression" box and that the operator is correct; in this case, it should show

```
((X,Y) in CIRCLE(25705.5,18820.5,1200))&&!((X,Y) in CIRCLE(25705.5,18820.5,400))
```

6) Follow steps 6 - 10 of §7.4.1 (GUI instructions), giving the output files appropriate names. In this case, let the spectrum be `mos1_bkg_pi.fits` and the filtered file be `mos1_filt_time_bkg.fits`.

Once again, the counts per channel are displayed automatically in *fv*.

To extract the background spectrum using the Command Window:

1) Follow steps 1 and 2 of §7.4.1 (Command Window instructions).

2) Make an annulus around the object whose background spectrum you wish to extract. This can be done using two circles, each defining the inner and outer edges of the annulus. Select the inner circle. From "Regions" on the pulldown menu, select "Properties", then "Exclude".

3) Adjust the sizes and positions of the circles until you are satisfied with the extraction region. In this case, we will center the annulus on the spectrum extraction region from §7.4.1 and let the background region extend from 20" to 1' around the source. Double-clicking on the regions will bring up windows listing the center coordinates and the radius of each circle.

4) Follow step 5 in §7.4.1 (Command Window instructions), remembering to change the output file names and the extraction expression:

```
spectrumset='mos1_bkg_pi.fits'
filteredset='mos1_filt_time_bkg.fits'
expression='((X,Y) in CIRCLE(25705.5,18820.5,1200))&&!((X,Y) in CIRCLE(25705.5,18820.5,400))'
```

   The spectrum, in counts per channel, may be seen with *fv*.

As with the source spectrum, the background spectrum's area needs to be set.

To find the background spectrum's source area using the GUI:

1) Call *backscale* in the Available Tools panel and enter the spectrum, `mos1_bkg_pi.fits` as the input data set.

2) Directly beneath, enter the file containing the bad pixel extensions, i.e., the event file from which it was extracted, `mos1_filt_time.fits`. Verify that the option to correct for bad pixels in the source box is set to "yes".

3) Click "Run".

To find the background spectrum's source area using the Command Window, type:

```
backscale spectrumset=mos1_bkg_pi.fits
   withbadpixcorr=yes badpixlocation=mos1_filt_time.fits
```

7.4.3 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. If this happens sufficiently often it will skew the spectrum to higher energies. To check whether pile up may be a problem, use the SAS task *epatplot*. Note that this procedure requires as input the event files created when the spectrum was made.

The output of *epatplot* is a postscript file, which when downloaded to the user's machine, may be viewed with *gv*. It contains two graphs describing the distribution of counts as a function of PI channel; see Figure 7.4. The top graph is the distribution of counts versus PI channel for each pattern class (single, double, triple, quadruple), and the bottom is the expected pattern distribution (**smooth lines**) plotted over the observed distribution (**histogram**). If the lower plot shows the model distributions for single and double events diverging significantly from the observed distributions, then the source is piled up.

The source used in this example is too faint to provide reasonable statistics for *epatplot* and is far from being affected by pile up. In contrast, Figure 7.5 shows an example of a bright source (from a different observation) which is strongly affected by pileup. Note the severe divergence between the model and the observed pattern distribution.

To check for pile up with the GUI:

1) In the Remote Directory List, highlight the name of the filtered event list that was made when the spectrum was extracted, `mos1_filt_time_source.fits`. In the the Available Tools window, call *epatplot*.

2) In the *epatplot* window, confirm that the event file is in the "Name of input events file" text box. Next to "Whether to determine background subtracted pattern fractions", click "yes", and in the text box just above, enter the name of the background spectrum event list (`mos1_filt_time_bkg.fits`). Enter the name of the output file; for this example, we will use `mos1_pat.ps`. Directly below, click "yes" next to "Use plotfile name from parameter".

3) Click "Run".

To check for pile up with the Command Window, type:

```
epatplot set=mos1_filt_time_source.fits plotfile=mos1_pat.ps
   useplotfile=yes withbackgroundset=yes backgroundset=mos1_filt_time_bkg.fits
```

The postscript file can be copied to the user's local machine and viewed there.

7.4.4 My Observation is Piled Up! Now What?

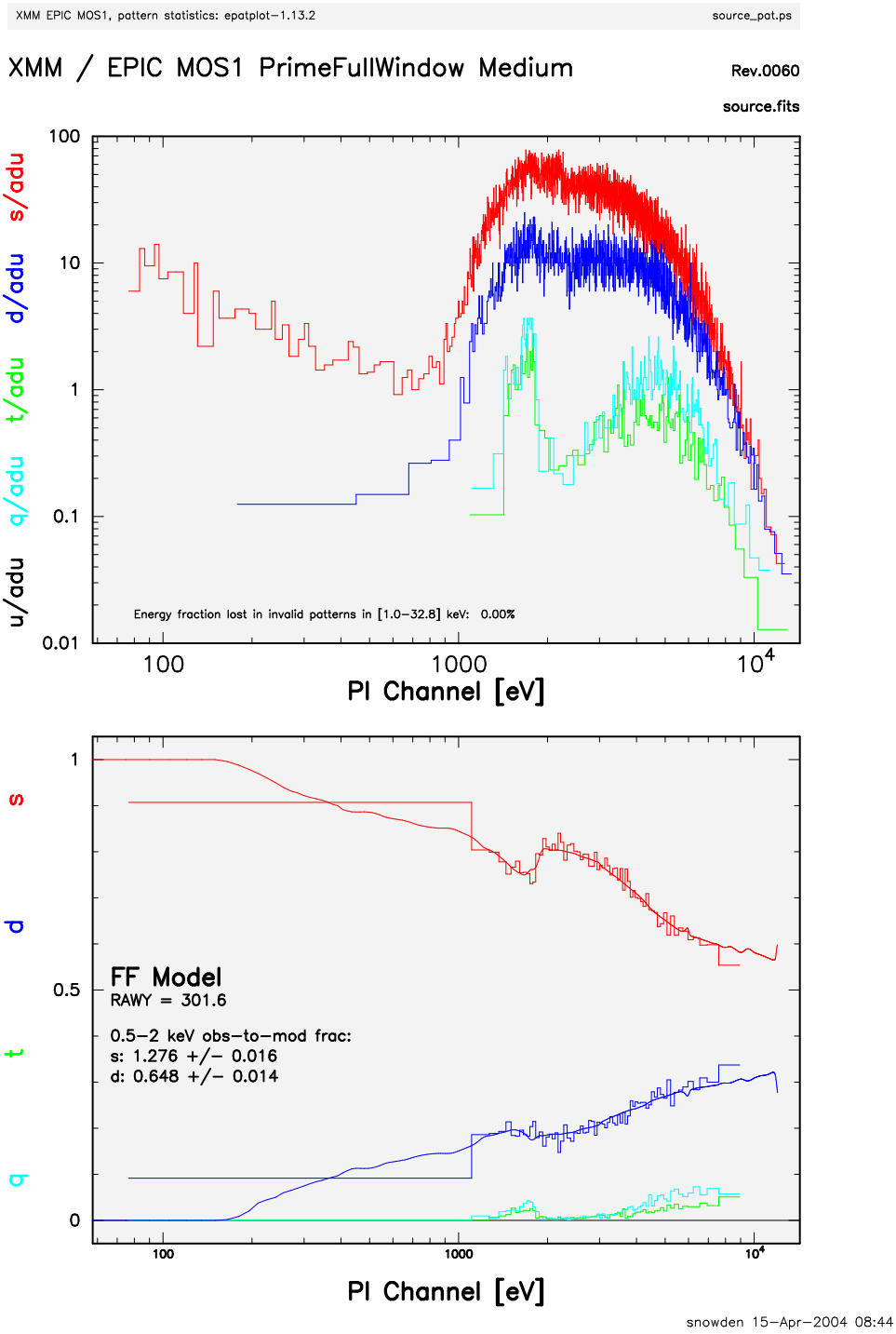
There are two ways to remove the effects of pile up from an observation:

- Using the region selection and event file filtering procedures demonstrated in earlier sections, you can excise the inner-most regions of a source (as they are the most heavily piled up), re-extract the spectrum, and continue your analysis on the excised event file. For this procedure, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with *epatplot*, and repeat, each time removing a slightly larger region, until the model and observed distribution functions agree.

and/or

- Using the event file filtering procedures, consider only the pattern 0 events (`PATTERN==0`). Pattern 0 events are less sensitive to pile up than other patterns.

Figure 7.4: The output of *epatplot* for a very faint source without pileup. Note that in the lower plot, for energies less than $\sim$ 1500 eV, there are too few X-rays for *epatplot* to model.



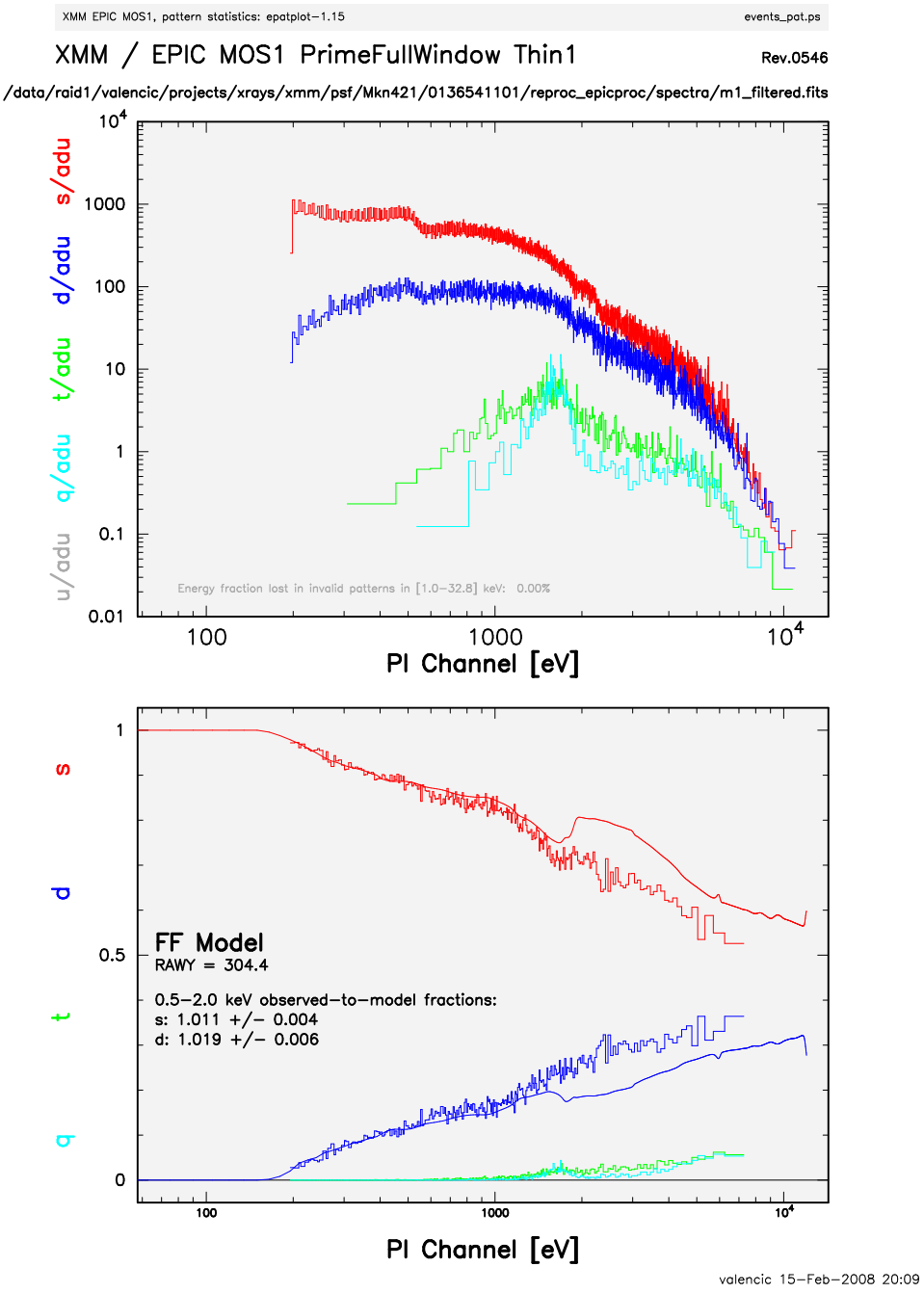
7.4.5 Create the Photon Redistribution Matrix (RMF)

In order to do spectral analysis, it is necessary to find the instrument's repsonse as a function of energy and PI channel. This is done by reformating the detector response and energy bounds information and correcting for instrumental effects, and writing the result to the Redistribution Matrix File (RMF). This is accomplished with the task *rmfgen*.

To make the RMF using the GUI:

1) Call *rmfgen* from the Available Tools panel.

Figure 7.5: The output of *epatplot* for a heavily piled source. In the lower plot, there are large differences between the predicted and observed pattern distribution at energies above $\sim$ 1000 eV.



2) Enter the name of the output dataset, in this case, `mos1_rmf.fits`. In the "Counts spectrum file" text box, enter the spectrum file name, in this case, `mos1_source_pi.fits`.

3) Click "Run".

To make the RMF using the Command Window, type:

```
rmfgen rmfset=mos1_rmf.fits spectrumset=mos1_source_pi.fits
```

where

`rmfset` – output file

`spectrumset` – spectrum file

7.4.6 Create the Ancillary Region File (ARF)

The Ancillary Response File (ARF) can be used with the RMF to fit spectral models to the observed spectrum. It contains a table of area values for several different energy ranges, having taken into account the calibration information and the state of the data.

To make the ARF using the GUI:

1) Call *arfgen* in the Available Tools panel.
2) Enter the name of the input spectrum, in this case, `mos1_source_pi.fits`, the input response set, `mos1_rmf.fits`, and the output ancillary response set, in this case, `mos1_arf.fits`.
3) Confirm that "Correct for bad pixels in source box" is set to "yes", and next to "File containing bad pixel extensions", enter the name of the event file from which the spectrum was extracted, in this case, `mos1_filt_time.fits`. Next to "Get ebins from response set", click "yes".
4) Click on "Run".

To make the ARF using the Command Window, type:

```
arfgen arfset=mos1_arf.fits spectrumset=mos1_source_pi.fits withrmfset=yes
   rmfset=mos1_rmf.fits badpixlocation=mos1_filt_time.fits
```

where

`arfset` – output ARF file name
`spectrumset` – input spectrum file name
`withrmfset` – flag to use the RMF
`rmfset` – RMF file created by *rmfgen*
`withbadpixcorr` – flag to include the bad pixel correction
`badpixlocation` – file containing the bad pixel information; should be set to the event file from which the spectrum was extracted.

7.4.7 Prepare the Spectrum

Assuming that source and background spectra have been extracted as in §7.4.1 and 7.4.2, and the RMF and ARF created as in §7.4.5 and 7.4.6, spectral fitting will be demonstrated using HEASoft software.

Nearly all spectra will need to be binned for statistical purposes. The procedure *grppha*, located in the HEASARC folder in the Available Tools window, provides an excellent mechanism to do just that. While some users may prefer the Hera GUI, it is often easier and more useful to simply enter the commands in the Command Window, especially as only one command may be entered at a time using the GUI interface. (With the GUI, the commands are entered in the two "GRPPHA" text boxes, and the last command entered must be `exit` to avoid a hung session.) In light of this, only instructions for the Command Window are shown here.

The following commands not only group the source spectrum for Xspec but also associate the appropriate background and response files for the source.

To group the source spectrum using the Command Window, type:

```
grppha
```

and edit the parameters and file names as appropriate:

```
Please enter PHA filename[] mos1_source_pi.fits ! input spectrum file name
Please enter output filename[] mos1_grp.fits    ! output grouped spectrum
GRPPHA[] chkey BACKFILE mos1_bkg_pi.fits        ! include the background spectrum
GRPPHA[] chkey RESPFILE mos1_rmf.fits           ! include the RMF
GRPPHA[] chkey ANCRFILE mos1_arf.fits           ! include the ARF
```

```
GRPPHA[] group min 25                          ! group the data by 25 counts/bin
GRPPHA[] exit
```

Upon exiting, the output file `mos1_grp.fits` will appear in your working directory.

7.4.8 Fit the Spectrum

Next, use Xspec to fit the spectrum. Again, since we will want to issue multiple commands, Hera GUI users are urged to use the Command Window.

To group the source spectrum using the Command Window, type:

```
xspec
```

A POW window will pop up and display the spectrum later on. Edit the parameters and file names as appropriate:

```
XSPEC> data mos1_grp.fits      ! input data
XSPEC> ignore 0.0-0.2,6.6-**   ! ignore unusable energy ranges, in keV
                               ! set a range appropriate for the data
XSPEC> model wabs(pow+pow)     ! set spectral model to two absorbed power laws
1:wabs:nH> 0.01                ! set model absorption column density to 1.e20
2:powerlaw:PhoIndex> 2.0       ! set the first model power law index to -2.0
3:powerlaw:norm>               ! use the default model normalization
4:powerlaw:PhoIndex> 1.0       ! set the second model power law index to -1.0
5:powerlaw:norm>               ! use the default model normalization
renorm                         ! renormalize the model spectrum
XSPEC> fit                     ! fit the model to the data
XSPEC> setplot energy          ! plot energy along the X axis
XSPEC> plot ldata ratio        ! plot two panels with the log of the data and
                               !     the data/model ratio values along the Y axes
XSPEC> exit                    ! exit Xspec
```

Figure 7.6 shows the fit to the spectrum.

7.5 Source Detection

The task *edetect_chain* is a metatask that does nearly all the work involved with EPIC source detection. It is comprised of seven straightforward tasks that can also be run by hand. *Edetect_chain* requires input files to be generated and prepared using the tasks *atthkgen* and *evselect*; the task *emosaic*, while not necessary for source detection, does provide a nice mosaicked image for display purposes. Fortunately, these are all quick and straightforward.

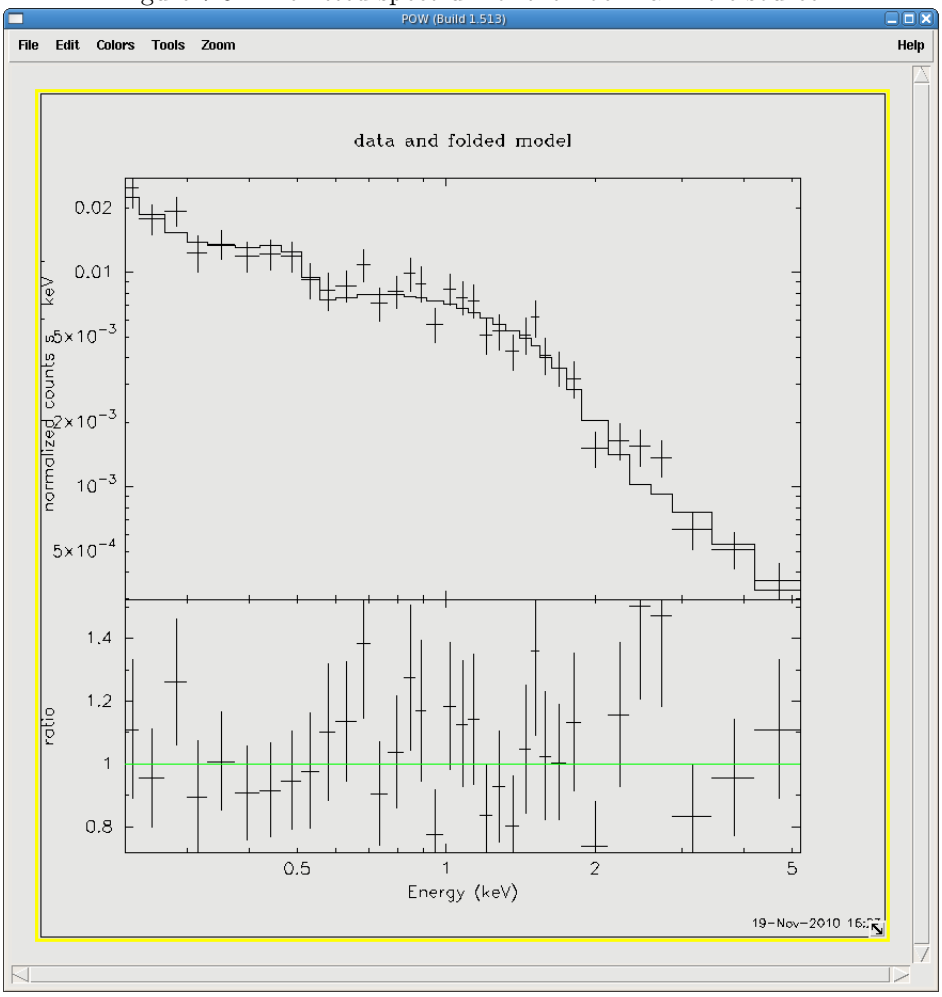
In the example below, source detection is done on images in two bands (300 - 2000 eV, and 2000 - 10000 eV, or "soft" and "hard", respectively) for all three detectors. Users interested in other bands might wish to look at Table 3.2 of the 2XMM Catalogue, which lists several bands and their ECFs (assuming an absorbed power law spectrum, with $N_H = 3.0 \times 10^{20}$ cm^{-2} and a photon index of $\Gamma = 1.7$.)

The example uses the filtered event file produced in §7.3.4.

To detect sources with the GUI:

1) Call *atthkgen* in the Available Tools panel.
2) Verify the name of the working directory, and enter the desired output file name, for example, `attitude.fits`.
3) Click "Run".

Figure 7.6: The fitted spectrum of the Lockman Hole source.



4) Select the event file (`mos1_filt_time.fits`) and run *heraXmmselect*.

5) In the "Selection Expression" box, enter our filtering parameters: `(FLAG == 0)&&(PI in [300:2000])`. Check the X and Y boxes, and click "Image".

6) In the "Image" tab, check the `withimageset` box, and enter the desired output image name, for example, `mos1-s.fits`. Set `xcolumn` to X and `ycolumn` to Y. Set `binning` to `binSize`, `ximagebinsize` to 22, and `yimagebinsize` to 22 for the MOS; for the PN, these should be set to 82.

7) Click "Run".

8) Repeat steps 4 - 7 for each event file, changing the output image name to something appropriate, for example, `mos2-s.fits` and `pn-s.fits`.

9) Set the "Selection Expression" text to `(FLAG == 0)&&(PI in [2000:10000])` and repeat steps 6 - 8. We will assume the output images are named `mos1-h.fits`, `mos2-h.fits`, and `pn-h.fits`, so that we now have six output images in all.

10) Call the task *emosaic* (the mosaicked image can be used for display purposes later; see Figure 7.7). Enter the input images: `mos1-s.fits mos1-h.fits mos2-s.fits mos2-h.fits`. Enter the output mosaicked image file name, for example, `mosaic.fits`.

11) Click "Run".

12) Call the task *edetect_chain*. Enter the list of images: `mos1-s.fits mos1-h.fits mos2-s.fits mos2-h.fits pn-s.fits pn-h.fits`. Directly below, enter the names of the event files: `mos1_filt_time.fits mos2_filt_time.fits pn_filt_time.fits`. Enter the name of the attitude file made by the task *atthkgen* (`attitude.fits`). Set the lower PI values (in eV) for the input images by typing: `300 2000 300 2000 300 2000`, and do similar for the higher values (`2000 10000 2000 10000 2000 10000`). The energy conversion factors (ECFs) for this example are `0.878 0.220 0.878 0.220 3.652 0.632`. The ECFs convert the source count rates into fluxes; for each detector and energy band, the ECFs depend on the pattern selection and filter used during the observation. For more information, please consult the calibration paper "SSC-LUX-TN-0059", available at the XMM-Newton Science Operations Center or see Table 3.2 in the 2XMM Catalogue User Guide. Those used here are derived from PIMMS using the flux in the 0.1-10.0 keV band, a source power-law index of 1.9, an absorption of 0.5×10^{20}. Set the flag to use out-of-time events in *esplinemap* to `no`.

13) Click "Run".

To detect sources in the Command Window:

1) Make the attitude file by typing

```
atthkgen atthkset=attitude.fits timestep=1
```

where

`atthkset` – output file name
`timestep` – time step in seconds for attitude file

2) Now make the event files with *evselect*.

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-s.fits
   imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50
   filtertype=expression expression='(FLAG == 0)&&(PI in [300:2000])'
```

where

`table` – event list
`withimageset` – flag to create an image
`imageset` – fits image name to be created
`imagebinning` – how to bin the image
`xcolumn` – table column to use for the X axis
`ximagebinsize` – binning in X axis
`ycolumn` – table column to use for the Y axis
`yimagebinsize` – binning in Y axis
`filtertype` – type of filtering
`expression` – filtering expression, select events in the PI channel range 300-2000 eV

3) Repeat step 2 for each detector, changing the output file name as needed. We will assume here that they are named `mos2-s.fits` and `pn-s.fits`.

4) Repeat steps 2 and 3, but change the filtering expression to `(FLAG == 0)&& (PI in [2000:10000])`. We will assume the output images are named `mos1-h.fits`, `mos2-h.fits`, and `pn-h.fits`, so that we now have six output images in all.

5) Create a merged count image to display later (see Figure 7.7):

```
emosaic imagesets='mos1-s.fits mos1-h.fits mos2-s.fits mos2-h.fits pn-h.fits pn-s.fits'
   mosaicedset=mosaic.fits
```

where

`imagesets` – list of count images
`mosaicedset` – output image name

6) Run *edetect_chain*.

```
edetect_chain imagesets='mos1-s.fits mos1-h.fits mos2-s.fits
   mos2-h.fits pn-s.fits pn-h.fits' eventsets='mos1_filt_time.fits
   mos2_filt_time.fits pn_filt_time.fits' attitudeset=attitude.fits
   pimin='300 2000 300 2000 300 2000' pimax='2000 10000 2000 10000 2000 10000'
   likemin=10 witheexpmap=yes ecf='0.878 0.220 0.878 0.220 3.652 0.632'
   eboxl_list=eboxlist_l.fits eboxm_list=eboxlist_m.fits
   eml_list=emllist.fits esp_withootset=no
```

where

`imagesets` – list of count images
`eventsets` – list of event files
`attitudeset` – attitude file name
`pimin` – list of minimum PI channels for the bands
`pimax` – list of maximum PI channels for the bands
`likemin` – maximum likelihood threshold
`witheexpmap` – create and use exposure maps
`ecf` – energy conversion factors for the bands
`eboxl_list` – output file name for the local sliding box source detection list
`eboxm_list` – output file name for the sliding box source detection in background map mode list
`eml_list` – output file name for maximum likelihood source detection list
`esp_withootset` – Flag to use an out-of-time processed PN event file, useful in cases where bright point sources have left streaks in the PN data
`esp_ooteventset` – The out-of-time processed PN event file

The energy conversion values (ECFs) can be suppled to convert the source count rates into fluxes. The ECFs for each detector and energy band depend on the pattern selection and filter used during the observation. For more information, please consult the calibration paper "SSC-LUX-TN-0059", available at the XMM-Newton Science Operations Center or see Table 3.2 in the 2XMM Catalogue User Guide. Those used here are derived from PIMMS using the flux in the 0.1-10.0 keV band, a source power-law index of 1.9, an absorption of 0.5×10^{20}.

7.6 Timing Analysis

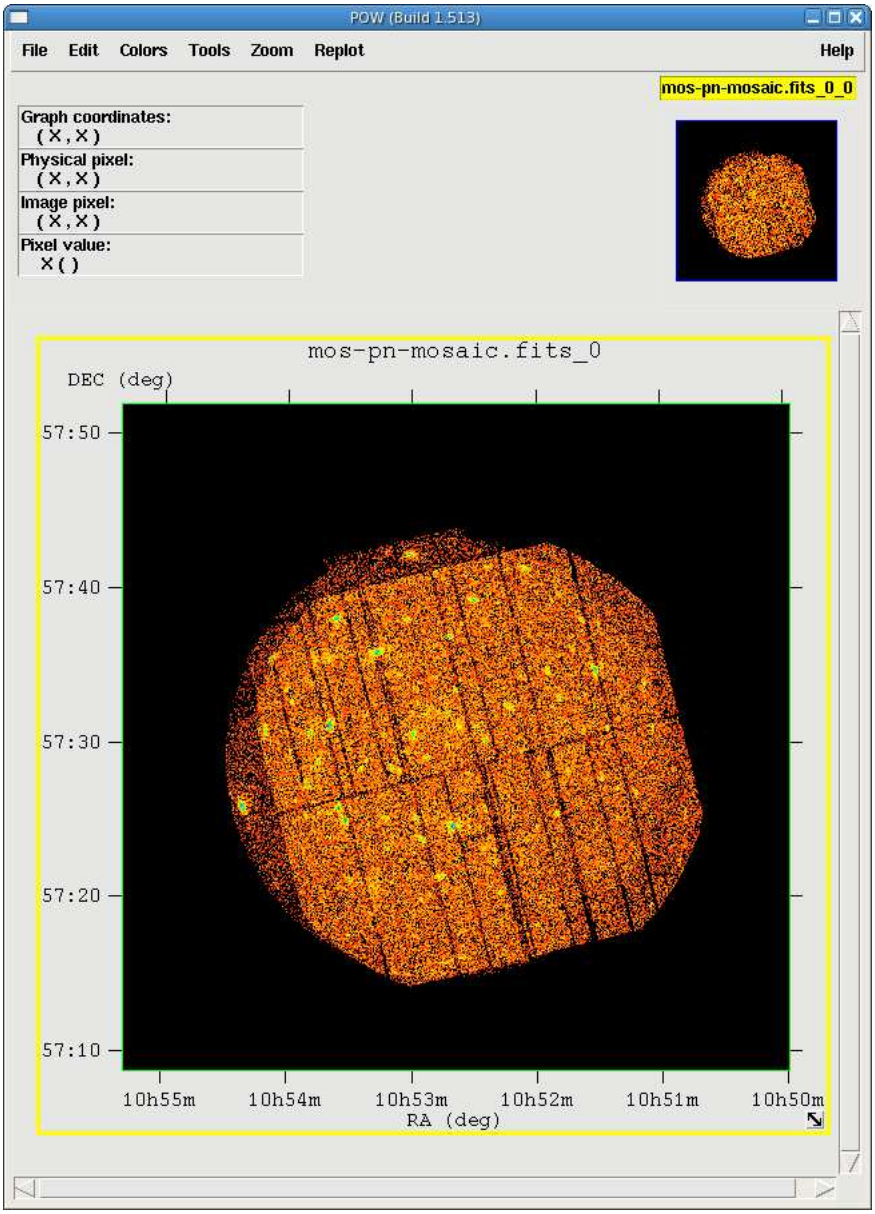
This section will demonstrate some basic timing analysis of EPIC image-mode data using the Xronos analysis package. For this exercise, the central source from the observation of G21.5-09 (Obs ID 0122700101) is used. These examples assume that the source's lightcurve has been made as in § 7.3.2, but with `timebinsize` set to `1` and `makeratecolumn` set to `no`; the name of this file assumed to be `source_ltcrv.fits`.
For the aficionado, the task *barycen* can be used for the barycentric correction of the source event arrival times. The Xronos tools can be access through the Hera GUI by clicking on XRONOS in the Available Tools panel.

To make a binned lightcurve with the Xronos GUI:

1) Call *lcurve* from the Available Tools panel.

2) Enter the Series 1 filename, in this case `source_ltcrv.fits`. Set "Name of the window file" to '-' (no quotes), "Newbin Time or negative rebinning" to 500, and the "Number of Newbins/Interval" to 430. Enter the name of the output file; for this example we will call it `lightcurve_binned.fits`. Next to "Do you want to plot your results?", select "No".

Figure 7.7: The mosaicked EPIC image created by *emosaic*, viewed with *fv*.



3) Click "Run".

The output can be viewed with *fv* by right-clicking on the filename and selecting the "Edit/Display File" option. Output is shown in Figure 7.8.

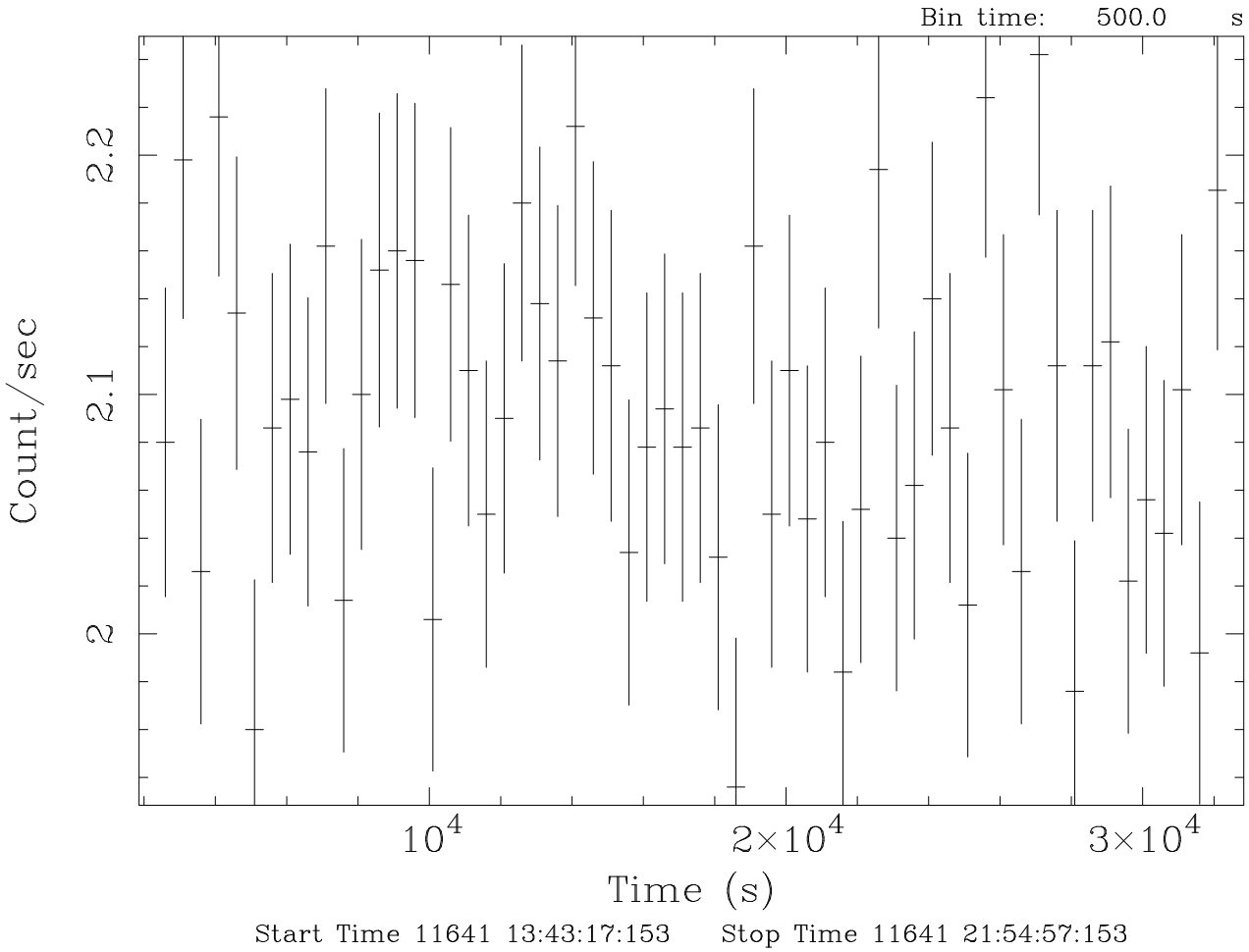
To make a binned lightcurve with Xronos in the Command Window:

```
lcurve nser=1 cfile1='source_ltcrv.fits' window=- dtnb=500 nbint=450
   outfile='lightcurve_binned.fits' plot=no
```

where

`nser` – number of time series
`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`dtnb` – bin size (time)

Figure 7.8: Light curve for the source analyzed as in § 7.3.2.



`nbint` – number of bins per interval
`outfile` – output file name (FITS format light curve)
`plot` – plot flag

The output can be viewed with *fv* by right-clicking on the filename and selecting the "Edit/Display File" option. Output is shown in Figure 7.8.

To calculate power spectrum density with the Xronos GUI:

1) Call *powspec* from the Available Tools panel.

2) Enter the Series 1 filename, in this case `source_ltcrv.fits`. Set "Name of the window file" to '-' (no quotes); next to "Newbin Time or negative rebinning", enter 100; for the number of newbins/interval, enter 300; for the number of intervals/frame, enter `INDEF`. Next to "Rebin results?", enter 5; next to "Do you want to plot your results?", select "No". Enter the name of the output file, in this case, `power.fits`.

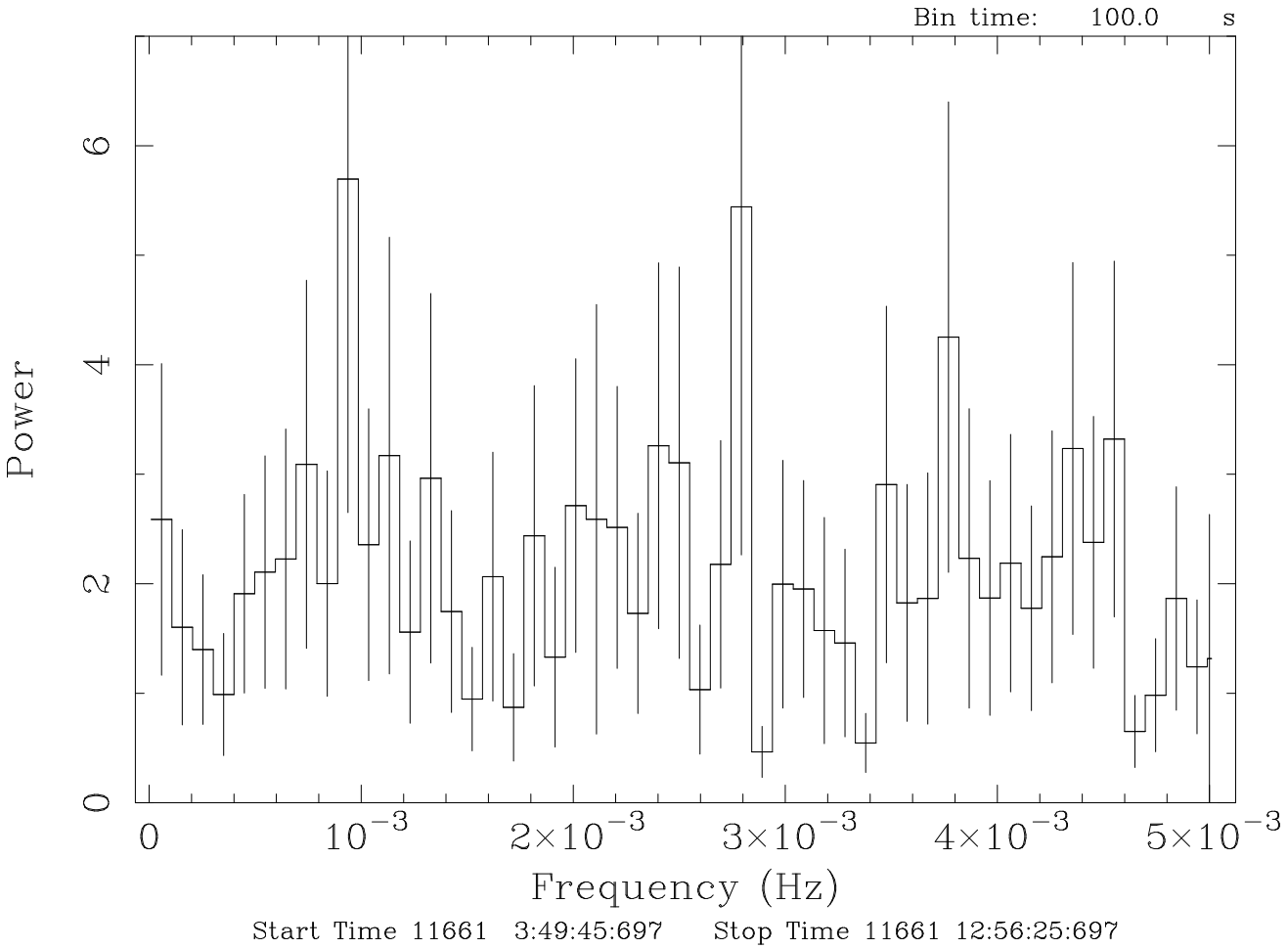
3) Click "Run".

The output can be viewed with *fv* by right-clicking on the filename and selecting the "Edit/Display File" option. Output is shown in Figure 7.9.

To calculate power spectrum density with Xronos in the Command Window, type:

```
powspec cfile1='source_ltcrv.fits' window=- dtnb=100.0 nbint=300
   nintfm=INDEF rebin=5 plot=no outfile='power.fits'
```

Figure 7.9: Power spectrum density for the source analyzed as in § 7.4.



where

`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`dtnb` – bin size (time)
`nbint` – number of bins per interval
`nintfm` – number of intervals in each power spectrum
`rebin` – rebin factor for power spectrum (0 for no rebinning)
`plot` – plot flag
`outfile` – output file name (FITS format power spectrum)

The output can be viewed with *fv* by right-clicking on the filename and selecting the "Edit/Display File" option. Output is shown in Figure 7.9.

To search for periodicities in the time series with the Xronos GUI:

1) Call *efsearch* from the Available Tools panel.

2) Enter the Series 1 filename, in this case `source_ltcrv.fits`. Set "Name of the window file" to '-' (no quotes); next to "Epoch", enter INDEF. Set "Period" to 20, "Phasebins/period" to 10, "Number of Newbins/Interval" to INDEF, "Number of periods to search" to 100, and "Resolution for period search" to INDEF. Next to "Do you want to plot your results?", select "No". Enter the name of the output file, in this case, `efsearch.fits`.

3) Click "Run".

To search for periodicities in the time series with Xronos in the Command Window, type:

```
efsearch cfile1=source_ltcrv.fits window=- sepoch=INDEF dper=20 nphase=10
   nbint=INDEF nper=100 dres=INDEF plot=no outfile=autocor.fits
```

where

`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`sepoch` – value for epoch used for phase zero when folding the time series
`dper` – value for the period used in the folding
`nphase` – number of phases per period
`nbint` – number of bins per interval
`nper` – number of sampled periods during search
`dres` – sampling resolution of search
`plot` – plot flag
`outfile` – output file name (FITS format)

To calculate the autocorrelation for a time series with the Xronos GUI:

1) Call *autocor* from the Available Tools panel.

2) Enter the Series 1 filename, in this case `source_ltcrv.fits`. Set "Name of the window file" to '-' (no quotes). Set "Newbin Time or negative rebinning" to 24, "Number of Newbins/Interval" to 2048; "Number of Intervals/Frame" to INDEF; and "Rebin results?" to 0. Next to "Do you want to plot your results?", select "No". Enter the name of the output file, in this case, `autocor.fits`.

3) Click "Run".

To calculate the autocorrelation for a time series with Xronos in the Command Window, type:

```
autocor cfile1=source_ltcrv.fits window=- dtnb=24.0 nbint=2048 nintfm=INDEF
   rebin=0 plot=no outfile=auto.fits
```

where

`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`dtnb` – bin size (time)
`nbint` – number of bins per interval
`nintfm` – number of intervals to be summed in each autocorrelation function
`rebin` – rebin factor for autocorrelation function (0 for no rebinning)
`plot` – plot flag
`outfile` – output file name (FITS format autocorrelation spectrum)

To calculate statistical quantities for a time series with the Xronos GUI:

1) Call *lcstats* from the Available Tools panel.

2) Enter the Series 1 filename, in this case `source_ltcrv.fits`. Set "Name of the window file" to '-' (no quotes). Set "Newbin Time or negative rebinning" to 6, and "Number of Newbins/Interval" to 8192.

3) Click "Run".

The statistics will be printed in the Command Window.

To calculate statistical quantities for a time series with Xronos in the Command Window, type:
write the output to an ASCII file `fname`. (Leave off the > `fname` to write the results to the screen.)

```
lcstats cfile1=source_ltcrv.fits window=- dtnb=6.0 nbint=8192
```

where

`cfile1` – filename first series
`window` – name of window file
`dtnb` – integration time (binning)
`nbint` – number of bins
`fname` – output file name

7.7 In a Nutshell

To summarize, the basic steps taken in EPIC data reduction are as follows.

1) Obtain the raw and pipelined data and store them on the Hera server.
2) Initialize Hera.
3) Make the calibration and ODF summary file (run the *cifbuild* and *odfingest* tasks).
4) Rerun the pipeline.
5) Apply standard filters.
6) Make light curves; re-filter the event files by time if necessary.
7) Extract the source and background spectra.
8) Check for pile up.
9) Make the response and ancillary files.

Chapter 8

An RGS Data Processing and Analysis Primer

While a variety of analysis packages can be used for the following steps, the SAS was designed for the basic reduction and analysis of XMM-Newton data and will therefore be used here for demonstration purposes.

As ever, it is strongly recommended that you keep all reprocessed data in its own directory! All flavors of Hera place output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipleline Processed data are in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, the reprocessing and analysis is taking place in the PROC directory. At this point, it is assumed that you have downloaded the data from the HEASARC archive onto a Hera server, standard or anonymous Hera is running (see §4.2), and you have prepared the data for processing with *cifbuild* and *odfingest* (see §6). Throughout this chapter, we will use the AB Dor dataset with ObsID 0134520301 available through links at the HEASARC archive.

8.1 A Quick Look at What You Have

The Pipeline Processed source lists for each detector are provided in zipped FITS format. For the RGS, these have the format `*R1*EVENLI*` and `*R2*EVENLI*`. These can be viewed through Hera with *fv* by double-clicking on the event file.

8.2 Rerunning the pipeline

It is very likely that you will want to filter your data to some extent; in this case, you will need to reprocess it in order to determine the appropriate filters, regardless of the age of the observation.

To rerun the pipeline:

1) Create the directory for processed data, PROC, by right-clicking on the ObsID directory and choosing the option "Create new directory". Highlight PROC by single-clicking on it.

2) In the Available Tools panel, under "XMM-SAS", call *rgsproc*. (The task automatically processes all exposures, so if you want to process data only from RGS1, set "enable explicit subset of exposures" to "yes", and directly below, in the "instrument exposure id list" box, enter the six-digit string that corresponds to the RGS1 data, e.g., "R1S001".) In the task pop-up window, click "Run".

3) Rename the output event file something easy to type, for example, `rgs1.fits`, by right-clicking on the repipelined event file. Throughout the following, we will assume that the processed RGS1 and RGS2 data are `rgs1.fits` and `rgs2.fits`, respectively.

To rerun the pipeline from the Command Window, type:

```
rgsproc
```

We will assume that the newly pipelined event files are named `rgs1.fits` and `rgs2.fits`.

Once the new event files have been obtained, the analysis techniques described later in this chapter can be used.

8.3 Potentially useful tips for using the pipeline

The pipeline task, *rgsproc*, is very flexible and can address potential pitfalls for RGS users. In §8.2, we used a simple set of parameters with the task, and if this is sufficient for your data, feel free to skip to §8.4. In the following sections, we will look at the cases of a nearby bright optical source, a nearby bright X-ray source, and a user-defined source.

8.3.1 A Nearby Bright Optical Source

With certain pointing angles, zeroth-order optical light may be reflected off the telescope optics and cast onto the RGS CCD detectors. If this falls on an extraction region, the current energy calibration will require a wavelength-dependent zero-offset. Stray light can be detected on RGS DIAGNOSTIC images taken before, during and after the observation. This test, and the offset correction, are not performed on the data before delivery.

To check for stray light and apply the appropriate offsets with the GUI:

1) Call *rgsproc*. Next to "calculate offsets from diagnostic mode" check "yes". Immediately below, next to "Store histograms in a dataset", check "no".

2) Click "Run".

8.3.2 A Nearby Bright X-ray Source

In the example above, it is assumed that the field around the source contains sky only. Provided a bright background source is well-separated from the target in the cross-dispersion direction, a mask can be created that excludes it from the background region. Here the source has been identified in the EPIC images and its coordinates have been taken from the EPIC source list which is included among the pipeline products. The bright neighboring object is found to be the third source listed in the sources file. The first source is the target.

To exclude a nearby source from the background region:

1) Double-click on the task *rgsproc* and confirm you are in the correct working directory. Next to "add sources from an epic-derived source list" click "yes". Immediately below, next to "name of the epic-derived source list", enter the name of the EPIC source list (as produced by such tasks as *emldetect* and *eboxdetect*). Next to "select which sources to exclude from the background", enter `INDEX==1&&INDEX==3`.

2) Click "Run".

8.3.3 User-defined Source Coordinates

If the true coordinates of an object are not included in the EPIC source list or the science proposal, the user can define the coordinates of a new source. For example, lets say that AB Dor wasn't the target of the proposal.

To define a new source's coordinates:

1) Double-click on the task *rgsproc* and confirm you are in the correct working directory. Next to "add a source with user-defined position, rate, etc" click "yes". Immediately below, next to "label for the user-defined source" enter the name of your source, `ABDor`. Next to "style for entering the position of the user-defined source", enter how the source position is to be defined; the choices are `radec` (celestial coordinates) and `wrtatt` (RGS angular coordinates). For `radec`, in the two boxes just below, enter the RA and declination of the source in decimal degrees; for `wrtatt`, in the two boxes below RA and dec, enter the dispersion-direction offset from the pointing axis and the cross-dispersion-direction offset from the pointing axis, each in arminutes. For this example, lets use `radec` and set the RA to `82.185493` and the declination to `-65.449329`.

2) Click "Run".

8.4 Examine and Filter the Data

Since the event files are current, we can proceed with some simple analysis demonstrations, which will allow us to generate filters. The following sections describe the use of Hera tasks using both the command line and GUI interfaces, except in cases where one of the methods is particularly easy. People new to Hera (or SAS) will likely prefer the GUI, at least at first; however, as they become more familiar with the software and the keywords, they will probably migrate to the command line, which is faster. Assuming that the parameter values for any given task are the same, it does not matter if a task is invoked on the command line or in the GUI; the output files will be identical. The *heraXmmselect* GUI, in the "experimental" folder of "XMM-SAS", provides a very simple method for producing and displaying images, spectra, and light curves, and is the recommended method for extracting data. An event list from the RGS1 can be loaded by highlighting the event file "rgs1.fits" in our PROC directory, then clicking on *heraXmmselect*, and "Run Tool...". A Table Selection window will pop up to confirm your selection; clicking "Go" will start *xmmselect*.

8.4.1 Create and Display an Image

Two commonly-made plots are those showing `PI` vs. `BETA_CORR` (also known as "banana plots") and `XDSP_CORR` vs. `BETA_CORR`.

To create images by using the GUI:

1) Check the square boxes to the left of the "`BETA_CORR`" and "`PI`" entries.
2) Click on the "Image" button near the bottom of the page. This brings up the *evselect* GUI (Figure 6.3).
3) In the "Image" tab in the *evselect* GUI, confirm that the `withimageset` box is checked.
4) In the `imageset` box, change the output image name from `image.ds` to something descriptive, in this case, `pi_bc.fits`.
5) Click on the "Run" button on the lower left corner of the *evselect* GUI.

Different binnings and other selections can be invoked by accessing the "Image" tab at the top of the GUI. The default settings are reasonable, however, for a basic image. Similarly, plots can be made comparing `BETA_CORR` to `XDSP_CORR`. These two example plots can be seen in Figure 8.1.

To create images from the Command Window, type:

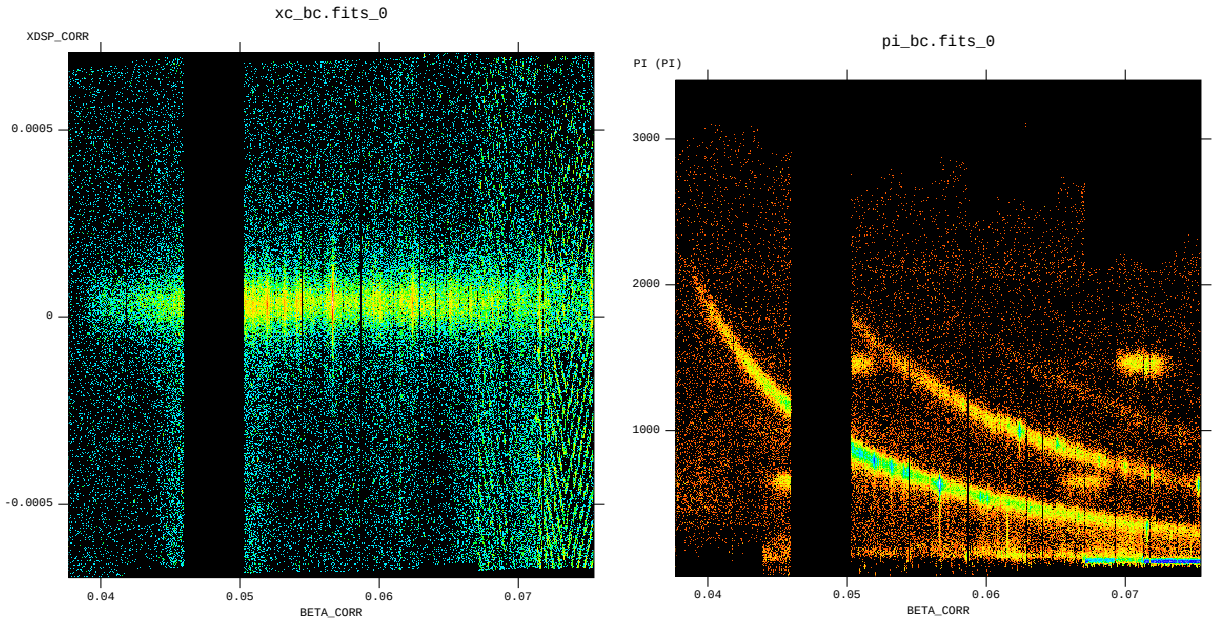
```
evselect table=rgs1.fits withimageset=yes
   imageset=pi_bc.fits xcolumn=BETA_CORR ycolumn=PI
   imagebinning=imageSize ximagesize=600 yimagesize=600
```

where

`table` – input event table
`withimageset` – make an image
`imageset` – name of output image
`xcolumn` – event column for X axis
`ycolumn` – event column for Y axis
`imagebinning` – form of binning, force entire image into a given size or bin by a specified number of pixels
`ximagesize` – output image pixels in X
`yimagesize` – output image pixels in Y

Plots comparing `BETA_CORR` to `XDSP_CORR` may be made in a similar way. The output files can be viewed with *fv*; example plot are shown in Figure 8.1.

Figure 8.1: Plots of XDSP_CORR vs. BETA_CORR (left) and PI vs. BETA_CORR (right). The gap is due to the missing CCD7. Similarly, CCD4 is missing in RGS2.



8.4.2 Create and Display a Light Curve

The background is assessed through examination of the light curve. We will extract a region, CCD9, that is most susceptible to proton events and generally records the least source events due to its location close to the optical axis. Also, to avoid confusing solar flares for source variability, a region filter that that removes the source from the final event list should be used. The region filters are kept in the source file product `P*SRCLI_*.FIT`. (For our example data, this would be `P0134520301R1S001SRCLI_0000.FIT`).

To create a light curve using the *heraXmmselect* GUI:

1) Enter the filtering criteria in the "Selection expression" box at the top of the *xmmselect* GUI:
`(CCDNR==9)&&(REGION(P0134520301R1S001SRCLI_0000.FIT:RGS1_BACKGROUND,BETA_CORR,XDSP_CORR))`

2) Check the diamond to the left of the `time` entry.

3) Click on the "OGIP Rate Curve" button near the bottom of the page. This brings up the *evselect* GUI (Figure 5.3).

4) In the "Lightcurve" tab, confirm that the `withrateset` box is checked. Change the `timebinsize` to a reasonable amount, e.g. 10 or 100 s, and change the default output file name in the `rateset` box to something appropriate, in this case, `r1_ltcrv.fits`.

5) Click on the "Run" button at the lower left corner of the *evselect* GUI.

   The resultant light curve is displayed automatically using POWplot (see Figure 8.2).

To create a light curve using the Command Window, type:

```
evselect table=rgs1.fits withrateset=yes rateset=r1_ltcrv.fits
   maketimecolumn=yes timebinsize=100 makeratecolumn=yes
   expression=
   '(CCDNR==9)&&(REGION(P0134520301R1S001SRCLI_0000.FIT:RGS1_BACKGROUND,BETA_CORR,XDSP_CORR))'
```

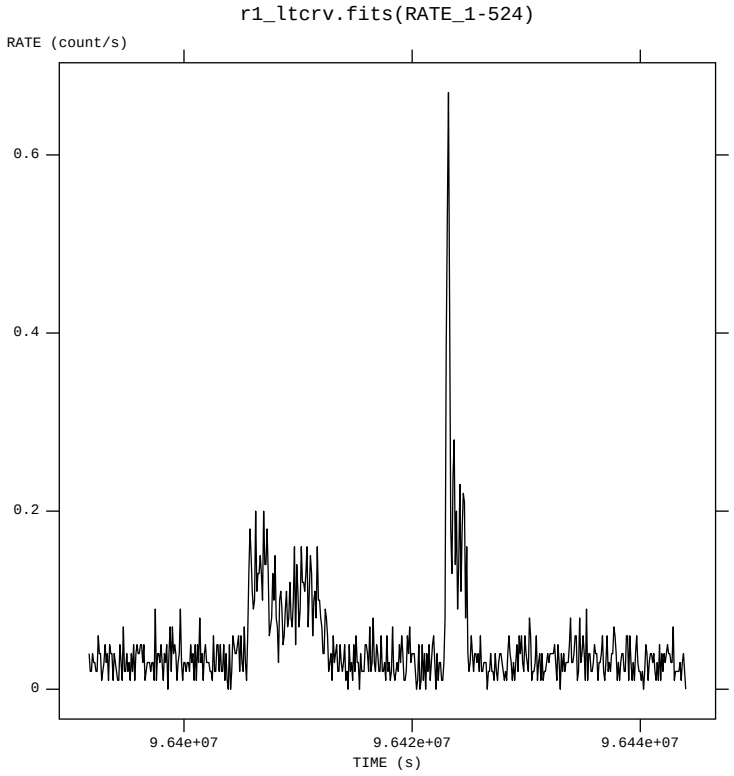
where

`table` – input event table
`withrateset` – make a light curve
`rateset` – name of output light curve file
`maketimecolumn` – control to create a time column
`timebinsize` – time binning (seconds)
`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

`expression` – filtering criteria

The output file `r1_ltcrv.fits` can be viewed with fv. The light curve is shown in Figure 8.2.

Figure 8.2: Background event rate from the RGS1 CCD9 chip. The flares are solar events. The time units are elapsed mission time.



8.4.3 Generating the Good Time Interval (GTI) File

Examination of the lightcurve shows that there are two noisy sections, one between 9.6405e7 and 9.6413e7 seconds, and another between 9.6422e7 and 9.6425e7 seconds. Both show rates well in excess of the normal background count rate of ~ 0.05 count/second. There are two procedures that make the GTI file (*gtibuild* and *tabgtigen*) that, when applied to the event file in another run of *rgsproc*, will excise these sections.

The first method, using *gtibuild*, requires a text file as input. This file can be made on your local machine and uploaded to your Hera account by right-clicking and dragging the file from your local directory to the remote directory. In the first two columns, refer to the start and end times (in seconds) that you are interested in, and in the third column, indicate with either a + or - sign whether that region should be kept or removed. In the example case, then, we would write in our ASCII file (named `r1_gti.txt`):

```
9.6405e7 9.6413e7 -
9.6422e7 9.6425e7 -
```

and proceed to the task *gtibuild*.

To make the GTI with *gtibuild* in the GUI:

1) Call *gtibuild* in the Hera GUI.
2) Enter the name of the text file and the output GTI table (here, `r1_gti.fits`).
3) Click "Run".

To make the GTI with *gtibuild* in the Command Window, type:

```
gtibuild file=r1_gti.txt table=r1_gti.fits
```

where

`file` – intput text file
`table` – output gti table

The task *tabgtigen* applies a filtering expression to the event file, limiting either the rates or the times, to produce the GTI.

To make the GTI with *tabgtigen* in the GUI:

1) Call the *tabgtigen* task.
2) Next to "Name of input data set", enter the name of the lightcurve file, `r1_ltcrv.fits`.
3) Directly below, enter the name of the output file, `r1_gti.fits`.
4) Next to "Booleam expression controlling the GTI creation", enter the filtering expression. Since the nominal count rate is about 0.05 count/sec, we will set the upper limit to 0.2 count/sec: `RATE<0.2`
5) Click "Run".

To make the GTI with *tabgtigen* in the Command Window, type:

```
tabgtigen table=r1_ltcrv.fits gtiset=r1_gti.fits expression='RATE<0.2'
```

where

`table` – the lightcurve file
`gtiset` – output gti table
`expression` – the filtering criteria. Since the nominal count rate is 0.05 about count/sec, we have set the upper limit to 0.2 count/sec.

8.4.4 Applying the GTI

Now that we have GTI file, we can apply it to the event file by running *rgsproc* again. *rgsproc* is a complex task, running several steps, with five different entry and exit points. It is not necessary to rerun all the steps in the procudure, only the ones involving filtering.

To rerun the pipeline in the GUI:

1) Call *rgsproc* in the Available Tools panel.
2) Next to "reflection orders for spectrum generation", make sure it is set for both orders, `1 2` .
3) Next to "(re)start processing at" and "conclude processing at", use the pull-down menus to select `3:filter` and `5:fluxing`, respectively.
4) Next to "Additional GTI tables", enter the name of the GTI file, `r1_gti.fits`.
5) Verify that "background correct the spectra" is set to `no`.
6) Verify that "Produce column with m*lambda (source at requested position) is set to "yes".

7) Click on "Run".

We will refer to the output file as `r1_filt.fits`.

To rerun the pipeline from the Command Window:

```
rgsproc orders='1 2' auxgtitables=r1_gti.fits bkgcorrect=no
   withmlambdacolumn=yes entrystage=3:filter finalstage=5:fluxing
```

where

`orders` – spectral orders to be processed
`auxgtitables` – gti file in FITS format
`bkgcorrect` – subtract background from source spectra?
`withmlambdacolumn` – include a wavelength column in the event file product
`entrystage` – stage at which to begin processing
`finalstage` – stage at which to end processing

We will refer to the output file as `r1_filt.fits`.

8.4.5 Creating the Response Matrices (RMFs)

Response matrices (RMFs) are not inlcuded in the Pipepline Processed package, and must be made prior to analyzing data. This can be done with the package *rgsrmfgen*. While these should only be used with point sources, they can be modified for use with extended sources; this is discussed in §8.7.2.

To make the RMFs using the GUI:

1) Call the *rgsrmfgen* task in Available Tools panel.
2) Next to "input events and exposure map extensions", enter the name of the filtered event list, `rgs1_filt.fits`. Next to "output response file", enter the name of the output file, `r1_o1_rmf.fits`. Next to "extracted spectrum file", enter the name of the spectrum file; it has the form `*SRSPEC*`, and in our case, is `P0134520301R1S001SRSPEC1001.FIT`. Next to "maximum output energy", enter 2.5; for the minimum output energy, enter 0.4. Next to "number of energy bins", enter 5000.
3) Click "Run".

To make the RMFs from the Command Window, type:

```
rgsrmfgen spectrumset=P0134520301R1S001SRSPEC1001.FIT rmfset=r1_o1_rmf.fits
   evlist=r1_filt.fits emin=0.4 emax=2.5 rows=5000
```

where

`spectrumset` – spectrum file
`evlist` – event file
`emin` – lower energy limit of the response file
`emax` – upper energy limit of the response file
`rows` – number of energy bins; this should be greater than 3000
`rmfset` – output FITS file

8.5 Fitting a Spectral Model

Now that we have a response file, we can fit the spectrum using Xspec. In the Command Window, type:

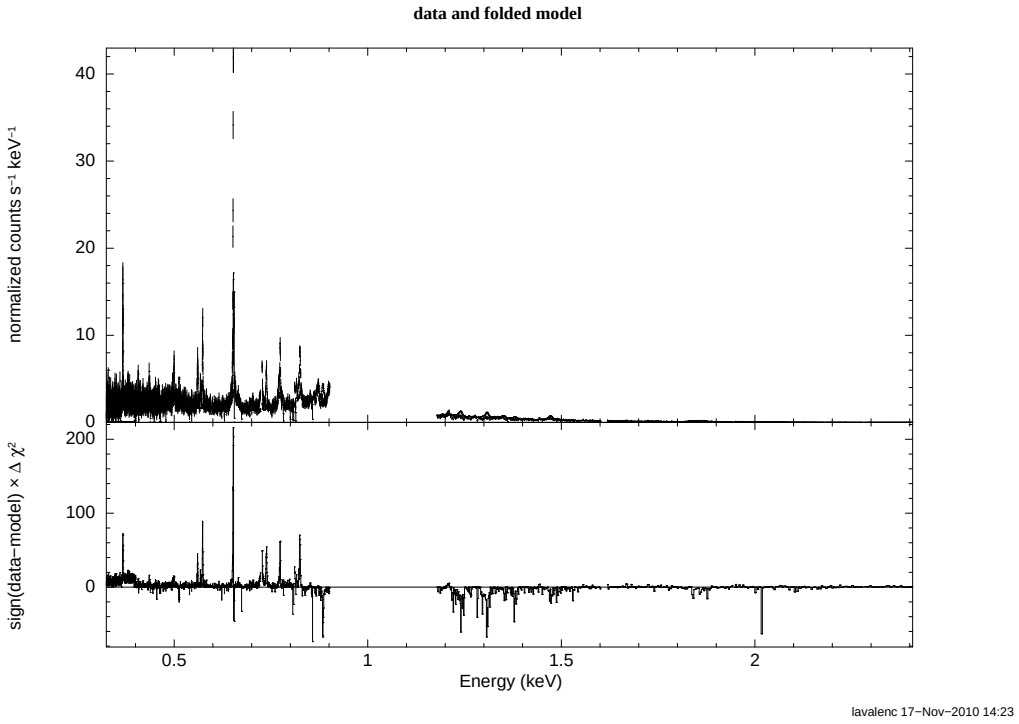
```
xspec
```

Enter the data, background, and response file at the prompts, and edit the fitting parameters as needed.

```
XSPEC> data P0134520301R1S001SRSPEC1001.FIT ! input data
XSPEC> back P0134520301R1S001BGSPEC1001.FIT ! input background
XSPEC> resp r1_o1_rmf.fits                  ! input response file
XSPEC> model wabs*mekal                     ! set spectral model to absorbed mekal
wabs:nH> 0.01
mekal:kT> 1
mekal:nH>
mekal:Anbundanc> .4
mekal:Redshift>
mekal:Switch> 0
mekal:norm> 1
XSPEC> renorm
XSPEC> fit 100
XSPEC> setplot energy
XSPEC> setplot command window all
XSPEC> setplot command log x off
XSPEC> setplot command wind 1
XSPEC> setplot command r y 0.01 40
XSPEC> plot data chisq
XSPEC> exit
```

Figure 8.3 shows the fit to the spectrum.

Figure 8.3: 1st order RGS1 spectrum of AB Dor. The fit is an absorbed single-temperature mekal model. The gap between 0.9 – 1.2 keV is due to the absence of CCD7.



8.5.1 Combining RGS1 and RGS2 Spectra

While it is tempting to merge the RGS1 and RGS2 data, or data from different pointings, to provide a single spectrum with a signal-to-noise improvement over either individual spectrum, this is strongly discouraged since

it results in data degradation.

The pointings of the two instruments are not identical, resulting in different dispersion angles and wavelength scales. Separate response files are always required for each unit. While it is possible to merge spectra and response files, great care must be taken to account for different exposure times, background subtractions, error propagation, and so on. However, the resulting response will always have inferior resolution to the originals. It is therefore simpler and more accurate to keep data from the two RGS units separate and use both sets to fit one model in tandem. Users who nonetheless wish to merge spectra from RGS1 and RGS2 can do so using the task *rgscombine*. Information on this task can be found here:
`http://xmm.esac.esa.int/sas/10.0.0/doc/rgscombine/index.html`

8.6 Approaches to Spectral Fitting

For data sets of high signal-to-noise and low background, where counting statistics are within the Gaussian regime, the data products above are suitable for analysis using the default fitting scheme in XSPEC, χ^2-minimization. However, for low count rates, in the Poisson regime, χ^2-minimization is no longer suitable. With low count rates in individual channels, the error per channel can dominate over the count rate. Since channels are weighted by the inverse-square of the errors during χ^2 model fitting, channels with the lowest count rates are given overly-large weights in the Poisson regime. Spectral continua are consequently often fit incorrectly, with the model lying underneath the true continuum level.

This will be a common problem with most RGS sources. Even if count rates are large, much of the flux from these sources can be contained within emission lines, rather than the continuum. Consequently, even obtaining correct equivalent widths for such sources is non-trivial. There are two approaches to fitting low signal-to-noise RGS data, spectral rebinning and maximum-likelihood statistics. The correct approach would normally be to use an optimization of the two.

8.6.1 Spectral Rebinning

By grouping channels in appropriately large numbers, the combined signal-to-noise of groups will jump into the Gaussian regime. There are two ways to do this: the FTOOL *grppha*, or the RGS pipeline. *grppha* can group channels using an algorithm which bins up consecutive channels until a count rate threshold is reached. This method conserves the resolution in emission lines above the threshold while improving statistics in the continuum.

To rebin with *grppha* in the Command Window, type:

```
grppha

        > Please enter PHA filename[] P0134520301R1S001SRSPEC1001.FIT
        > Please enter output filename[] P0134520301R1S001SRSPEC1001.bin.FIT
        > GRPPHA[] group min 30
        > GRPPHA[] exit
```

The disadvantage of using *grppha* is that, although channel errors are propagated through the binning process correctly, the errors column in the original spectrum product is not strictly accurate. The problem arises because there is no good way to treat the errors within channels containing no counts. To allow statistical fitting, these channels are arbitrarily given an error value of unity, which is subsequently propagated through the binning. Consequently, the errors are overestimated in the resulting spectra.

The other approach, which involves calling the RGS pipeline, bins the data during spectral extraction. The following rebins the pipeline spectrum by a factor 3. Note that the entry stage is much later in the *rgsproc* chain of processes, so this takes far less time than when run with an earlier entry stage.

To rebin with *rgsproc* in the GUI:

1) Call *rgsproc* in the "Available Tools" panel of the Hera GUI.

2) Next to "(re)start processing at", use the pull-down menu to select "4:spectra". Next to "dispersion channel rebinning factor", enter 3; next to "number of response energy bins", enter 4000; verify "background correct the spectra" is set to "no".

3) Click "Run".

To rebin with *rgsproc* in the Command Window, type:

```
rgsproc orders='1 2' rebin=3 rmfbins=4000 entrystage=4:spectra
   finalstage=5:fluxing bkgcorrect=no
```

where

`orders` – dispersion orders to extract
`rebin` – wavelength rebinning factor
`rmfbins` – number of bins in the response file; this should be greater than 3000
`entrystage` – entry stage to the pipeline
`finalstage` – exit stage for the pipeline

One disadvantage of this approach is that you can only choose integer binning of the original channel size. To change the sampling of the events, the pipeline must be run from the second stage ("angles") or earlier.

To change the event sampling with *rgsproc* in the GUI:

1) Call *rgsproc* in the "Available Tools" panel of the Hera GUI.

2) Next to "(re)start processing at", use the pull-down menu to select "2:angles". Next to "number of response energy bins", enter 4000; verify "background correct the spectra" is set to "no". Next to "number of beta bins", enter "1133".

3) Click "Run".

To change the event sampling with *rgsproc* in the Command Window, type:

```
rgsproc orders='1 2' nbetabins=1133 rmfbins=4000 entrystage=2:angles
   finalstage=5:fluxing bkgcorrect=no
```

where the parameters are as defined previously, and

`nbetabins` – number of bins in the dispersion direction; the default is 3400

The disadvantage of using *rgsproc*, as opposed to *grppha*, is that the binning is linear across the dispersion direction. Velocity resolution is lost in the lines, so the accuracy of redshift determinations will be degraded, transition edges will be smoothed, and neighboring lines will become blended.

8.6.2 Maximum-Likelihood Statistics

The second method is to replace the χ^2-minimization scheme with the Cash maximum-likelihood scheme (`cstat` in Xspec) when fitting data. This method is much better suited to data with low count rates and is a suitable option only if one is running Xspec v11.1.0 or later. The reason for this is that RGS spectrum files have prompted a slight modification to the OGIP standard. Because the RGS spatial extraction mask has a spatial-width which is a varying function of wavelength, it has become necessary to characterize the `BACKSCL` and `AREASCL` parameters as vectors (i.e., one number for each wavelength channel), rather than scalar keywords as they are for data from the EPIC cameras and past missions. These quantities map the size of the source extraction region to the size of the background extraction region and are essential for accurate fits.

One caveat of using the `cstat` option is that the scheme requires a "total" and "background" spectrum to be loaded into Xspec. This is in order to calculate parameter errors correctly.

To change schemes in Xspec before fitting the data in the Command Window, type:

```
xspec
statistic cstat
```

8.7 Analysis of Extended Sources

8.7.1 Region masks

The optics of the RGS allow spectroscopy of reasonably extended sources, up to a few arc minutes. The width of the spatial extraction mask is defined by the fraction of total events one wishes to extract. With the default pipeline parameter values, over 90% of events are extracted, assuming a point-like source.

Altering and optimizing the mask width for a spatially-extended source may take some trial and error, and, depending on the temperature distribution of the source, may depend on which lines one is currently interested in. While AB Dor is not an extended source, the following example increases the width of the extraction mask and ensures that the size of the background mask is reduced so that the two do not overlap.

To adjust the region mask with *rgsproc* in the GUI:

1) Call *rgsproc* in the "Available Tools" panel of the Hera GUI.

3) Next to "(re)start processing at", use the pull-down menu to select "4:spectra". Next to "percent of cross-dispersion PSF included" and "percent of cross-dispersion PSF excluded from the background", enter 99. Next to "percent of pulse-height distribution included", enter 95.

4) Click "Run".

To adjust the region mask with *rgsproc* in the Command Window:

```
rgsproc orders='1 2' entrystage=4:spectra finalstage=5:fluxing bkgcorrect=no
   xpsfincl=99 xpsfexcl=99 pdistincl=95
```

where parameters are as they were decsribed previously, and

`xpsfincl` – include this fraction of point-source events inside the spatial source extraction mask
`xpsfexcl` – exclude this fraction of point-source events from the spatial background extraction mask
`pdistincl` – include this fraction of point-source events inside the pulse height extraction mask

Observing extended sources effectively broadens the psf of the spectrum in the dispersion direction. Therefore, it is prudent to also increase the width of the PI masks using the `pdistincl` parameter in order to prevent event losses.

8.7.2 Making RMFs for extended sources

RGS response matrices as made in §8.4.5 are appropriate for use with point sources only. If we are interested in analyzing an extended source, the RMF must take into account the spatial degradation of the resolution. The most straight-forward way to do this is to modify the response matrix prior to spectral fitting. For sources extended up to about 1 arcminute, this can be done with the FTOOL *rgsrmfsmooth*. It requires three files: the point source RMF (as made in §8.4.5), an image of the source (from an EPIC camera, see §7.3.1, or different mission), and a text file. The better the resolution of the image, the more accurate the modified RMF will be, so if a Chandra image is available for a source, it should be used instead of an EPIC image. The text file must list the name of the image, the boresight, and the aperture size in the following format:

```
RGS_XSOURCE_IMAGE <name of source image>
RGS_XSOURCE_BORESIGHT <image boresight:  RA (h:m:s), DEC(degrees:m:s), and PA (decimal
   degrees)>
RGS_XSOURCE_EXTRACTION <source radius in arcminutes>
```

For an example case, we will name our text file `xsource.mod`. We will assume that a RMF for the first order grating was made as in §8.4.5 and an MOS1 image was made as in §7.3.1; `xsource.mod` contains these lines:

```
RGS_XSOURCE_IMAGE my_MOS1_image.fits
RGS_XSOURCE_BORESIGHT 05:28:45 -65:26:55 219.769546508789
RGS_XSOURCE_EXTRACTION 1.2
```

This file can be made on the user's local machine and uploaded to the Hera server by right-clicking and dragging the file from the Local Directory panel to the Remote Directory.

To make an RMF for an extended source in the Command Window, type:

```
rgsrmfsmooth rmffil=my_ps_rmf.fits imgfil=xsource.mod order=1 outfil=my_es_rmf.fits
```

where

`rmffil` – the point source RGS RMF
`imgfil` – the text file with the name of the source image, boresight, and extraction region
`order` – grating order of RMF
`outfil` – output RMF name

8.8 In A Nutshell

To summarize, the steps taken in RGS data reduction are as follows.

1) Obtain the raw and pipelined data and store them on the Hera server.
2) Initialize Hera.
3) Make the calibration and ODF summary file (run the *cifbuild* and *odfingest* tasks).
4) Rerun the pipeline.
5) Make a light curve; if necessary, filter the event files by time by rerunning the pipeline again.
6) Make the response file.

Chapter 9

An OM Data Processing and Analysis Primer

As with EPIC and RGS datasets, many files are associated with an OM dataset. The `INDEX.HTM` file, and links therein, are viewable with a web browser and will help you navigate the dataset. The different types of files are discussed in Chapter 5.2; however, since the OM is somewhat different from the other instruments on-board XMM-Newton, we will discuss them in more detail in later sections.

The OM can operate in IMAGING, FAST, and GRISM mode. Each of these modes has dedicated commands to reprocess the data: *omichain*, *omfchain*, and *omgchain*. These are Perl scripts which each call several procedures sequentially that are used to prepare the data for processing, make and apply flatfield images, and detect sources. The tasks *omichain* and *omfchain* also calculate the instrumental magnitudes of sources, find the position of the sources (in equatorial coordinates), and produce a sky image; *omgchain* produces a spectrum. If you run these chains, it is helpful to inspect the `sas_log` file to get a detailed list of the performed tasks. These chains rely on filters specified by the user; if no arguments are given, they run on all the files present in the ODF directory. **Due to the long file names and the large number of input parameters, users are urged to simply use the chains and not run the chains' individual tasks one at a time.**

Most OM data are obtained in IMAGING mode. If they were obtained in the FAST mode, there will be an additional event list file corresponding to the Fast window (`*FAE.FIT`). Reprocessing of data taken in FAST mode is discussed in §9.4. Reprocessing OM GRISM data is discussed in §9.5.

As always, it is **strongly** recommended that you keep all reprocessed data in its own directory! Hera places output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipleline Processed data in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, and the analysis is taking place in the PROC directory.

At this point, it is assumed that you have downloaded the data from the HEASARC archive onto a Hera server, standard or anonymous Hera is running (see §4.2), and you have prepared the data for processing (see Chapter 6). We will use the example data as noted in Table 6.1, though any dataset with the appropriate mode will suffice.

9.1 OM Artifacts and General Information

Before proceeding with the pipeline, it is appropriate to discuss the artifacts that often affect OM images. These can affect the accuracy of a measurement by, for example, increasing the background level. Some of these can be seen in Fig. 9.1.

- Stray light – background celestial light is reflected by the OM detector housing onto the center on the OM field of view, producing a circular area of high background. This can also produce looping structures and long streaks.
- Modulo 8 noise – In the raw images, a modulo 8 pattern arises from imperfections in the event centroiding algorithm in the OM electronics. This is removed during image processing.
- Smoke rings – light from bright sources is reflected from the entrance window back on the detector, producing faint rings located radially away from the center of the field of view.

- Out-of-time events – sources with count rates of several tens of counts/sec show a strip of events along the readout direction, corresponding to photons that arrived while the detector was being read out.

Further, artifacts also can contaminate grism data. Due to this mode's complexity, users are urged to be very careful when working with grism data, and should refer to the SOC's website on this topic.

Users should also keep in mind some differences between OM data and X-ray data. Unlike EPIC and RGS, there are no good time intervals (GTIs) in OM data; an entire exposure is either kept or rejected. Also, OM exposures only provide direct energy information when in grism mode, and the flat field response of the detector is assumed to be unity.

9.2 A Quick Look at What You Have

The PPS directory for the OM products contains files with nomenclature as described in Tables 5.2 and 5.3. As can be seen in those tables, the OM produces, among other things, sky images (`*SIMAGE*.FTZ`) and source lists (`*SWSRLI*.FTZ`). Further, there are low resolution sky images for each filter; they follow the nomenclature:

- `Pjjjjjjkkkk0MX000LSIMAGb000.QQQ`

  `jjjjjj` – Proposal number

  `kkkk` – Observation ID

  `b` – Filter keyword: B, V, U, M (UVM2), L (UVW1), and S (UVW2)

  `QQQ` – File type (e.g., PNG, FTZ)

So for example, `P0123700101OMX000RSIMAGV000.FTZ` is the final low resolution sky image in the V filter. As with all images, it can be viewed in Hera with *fv*. To see what files have been summed to make the final image, search for the keyword `XPROC0` in the FITS header. For our example image, this would be

```
XPROC0  = 'ommosaic imagesets=''product/P0123700101OMS004SIMAGE1000.FIT produc&'
CONTINUE  't/P0123700101OMS415SIMAGE1000.FIT product/P0123700101OMS416SIMAGE10&'
CONTINUE  '00.FIT product/P0123700101OMS417SIMAGE1000.FIT product/P0123700101O&'
CONTINUE  'MS418SIMAGE1000.FIT'' mosaicedset=product/P0123700101OMX000RSIMAGV0&'
CONTINUE  '00.FIT exposuremap=no exposure=1000 # (ommosaic-1.11.7) [xmmsas_200&'
CONTINUE  '61026_1802-6.6.0]'
```

The source list file (`*SWSRLI*.FTZ`) also contains useful information for the user; the column names are listed in Table 9.1.

Table 9.1: Some of the important columns in the source list file.

Column name	Contents
SRCNUM	Source number
RA	RA of the detected source
DEC	Dec of the detected source
POSERR	Positional uncertainty
RATE	extracted count rate
RATE_ERR	error estimate on the count rate
SIGNIFICANCE	Significance of the detection (in σ)
MAG	Brightness of the source in magnitudes
MAGERR	uncertainty on the magnitude

9.3 Imaging Mode

9.3.1 Rerunning the Pipeline

First, create the directory for processed data, PROC, by right-clicking on the ObsID directory and choosing the option "Create new directory". Highlight PROC by single-clicking on it. Please note that both the Hera GUI and Command Window rely on this to know where to place the output files. Calling any of the chains will initiate processing on all OM data of that particular mode; currently, only *omichain* will accept parameters to limit processing to a specific filter or exposure.

To rerun the pipeline on all exposures and filters from the GUI:

1) In the Available Tools panel, under "XMM-SAS", call *omichain*.

2) In the task pop-up window, click "Run".

To rerun the pipeline on all exposures and filters from the Command Window, type:

```
omichain
```

This produces numerous files, including images and regions for each exposure and each filter. If we are interested in the sources detected in the mosaicked, V band image we could run *omichain* with the appropriate flags.

To rerun the pipeline on the mosaicked, V-band image from the GUI:

1) In the Available Tools panel, under "XMM-SAS", call *omichain*, Next to "List of OM filters to be processed", enter `V`; next to "Source detection on stacked images", click "yes". Next to "Number of sigma for a detection", enter `2.0`; next "Minimum allowed source significance", enter `3.0`.

2) In the task pop-up window, click "Run".

To rerun the pipeline on the mosaicked, V-band image from the Command Window, type:

```
omichain filters=V processmosaicedimages=yes omdetectnsigma=2.0
   omdetectminsignificance=3.0
```

where

`filters` – list of filters to be processed
`processmosaicedimages` – process the mosaicked sky images?
`omdetectnsigma` – number of σ above background required for a pixel to be considered part of a source
`omdetectminsignificance` – minimum significance of a source to be included in the source list file

The output files can be used immediately for analysis, though users are strongly urged to examine the output for consistancy first (see §9.3.2). The chains apply all necessary corrections, so no further processing or filtering needs to be done. Please note that the chains do not produce output files with exactly the same names as those in the PPS directory (they also produce some files which are not included in the PPS directory at all.) Table 9.2 lists the file ID equivalences between repipelined and PPS files.

9.3.2 Verifying the Output

While the output from the chains is ready for analysis, OM does have some peculiarities, as discussed in §9.1. While these usually have only an aesthetic effect, they can also affect source brightness measurements, since they increase the background. **In light of this, users are strongly encouraged to verify the consistency of the data prior to analysis.** There are a few ways to do this. Users can examine the combined source list with *fv*, which will let them see if interesting sources have been detected in all the filters where they are visible. Users can also overlay the image source list on to the sky image with *implot* by downloading the relevant files to the user's local machine. This can also be done with *ds9* or *gaia* by using *slconv* to change source lists into region files and downloading the relevant files to your local machine. The task *slconv* allows users to set the regions radii in arcseconds to a constant value or scale them to header keywords, such as RATE. By default,

Table 9.2: File ID equivalences between repipelined and PPS OM files.

Repipelined Name	PPS Name	Description
EVLIST	none	FAST mode events list
FIMAG_	FIMAG_	combined full-frame image
FLAFLD	none	flatfield
FSIMAG	FSIMAG	combined full-frame sky image
HSIMAG	HSIMAG	full-frame HIRES sky image mosaic
IMAGE_	IMAGE_	image from any filter or GRISM
IMAGE_	IMAGEF	FAST mode image
LSIMAG	LSIMAG	full-frame LORES sky image mosaic
OBSMLI	OBSMLI	combined observation source list
REGION	SWSREG	sources region file
REGION	SFSREG	FAST mode sources region file
REGION	SGSREG	GRISM *ds9* regions
RIMAGE	GIMAGE	GRISM rotated image
RSIMAG	RSIMAG	default mode sky mosaic
SIMAGE	SIMAGE	sky aligned image
SIMAGE	SIMAGF	FAST mode sky aligned image
SIMAGE	none	GRISM sky aligned image
SPCREG	SPCREG	GRISM *ds9* spectrum regions
SPECLI	SPECLI	GRISM specra list
SPECTR	SPECTR	source extracted spectra
SUMMAR	SUMMAR	observation summary
SWSRLI	SWSRLI	sources list
SWSRLI	SFSRLI	FAST mode sources list
SWSRLI	SGSRLI	GRISM sources list
TIMESR	TIMESR	FAST mode source timeseries
TSHPLT	TSHPLT	tracking history plot
TSTRTS	TSTRTS	tracking star timeseries

ds9 region files have suffixes of `.reg`; *gaia* region files have suffixes of `.gaia`. In the example below, we make a region file from the source list for the mosaicked, V-band sky image.

To make a *ds9* region file from a source list with the GUI:

1) In the Available Tools panel, under "XMM-SAS", call *slconv*.
2) Next to "Name of source list dataset:table", enter `P0123700101OMS000RSISWSV.FIT:SRCLIST`; next to "Expression that determines the radius of the source circle", enter `5`; next to "Output mode", enter `ds9`; next to "First part of the name of the output text file", enter `Vband_mosaic`.
3) In the task pop-up window, click "Run".

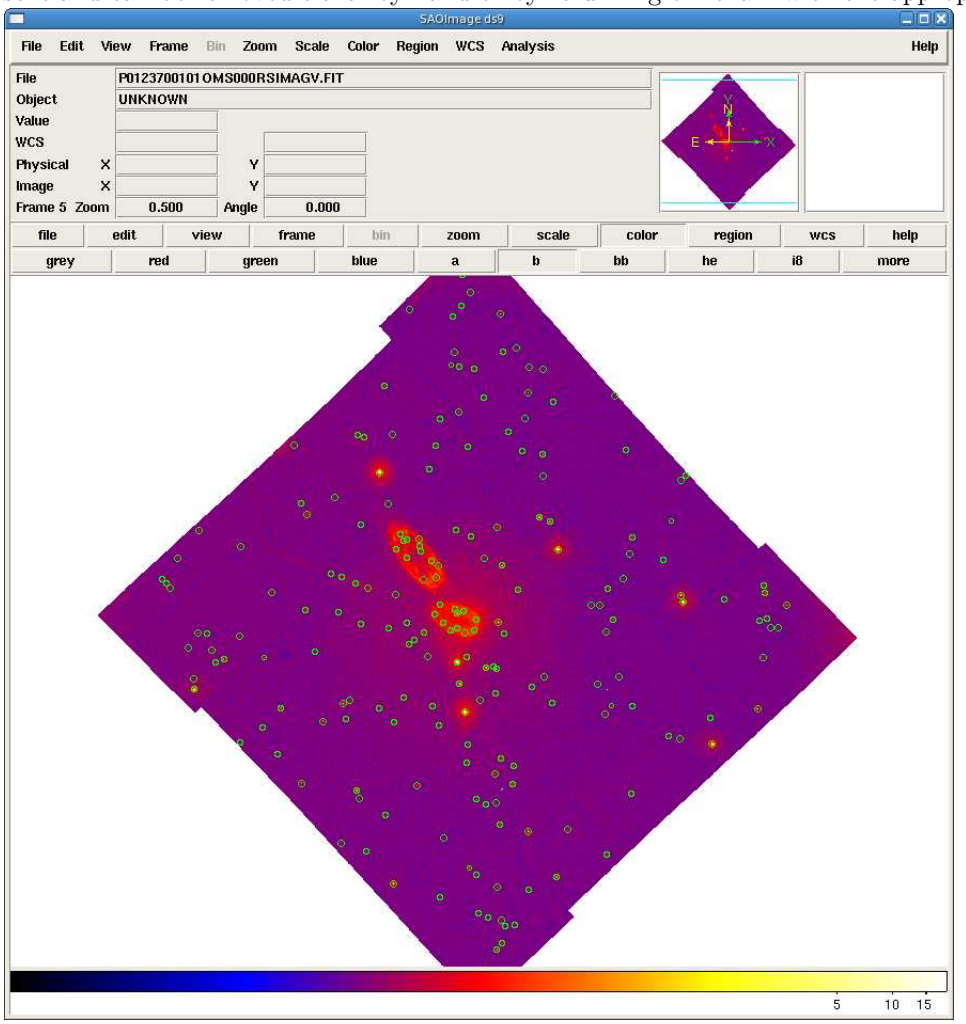
To make a *ds9* region file from a source list in the Command Window, type:

```
slconv srclisttab=P0123700101OMS000RSISWSV.FIT radiusexpression=5
   outfileprefix=Vband_mosaic outputstyle=ds9
```

where

`srclisttab` – source list file name
`radiusexpression` – constant or expression (possibly involving keywords) used

Figure 9.1: A mosaicked, V-band sky image and corresponding region file viewed with *ds9*. Spurious detections are clearly present and can be removed either by hand or by rerunning omichain with the appropriate flags.



to determine the radii of the plotted circles

`outputstyle` – output format; either ds9 or gaia

`outfileprefix` – prefix of output file name

We have downloaded the image and region file for examination on a local machine. The mosaicked, V-band sky image, `P0123700101OMS000RSIMAGV.FIT`, with the region file from *slconv* overlayed, is shown in Fig. 9.1. There clearly are spurious detections; these can be removed by hand, or by rerunning *omichain* with a different background-level threshold or source significance.

9.4 Fast Mode

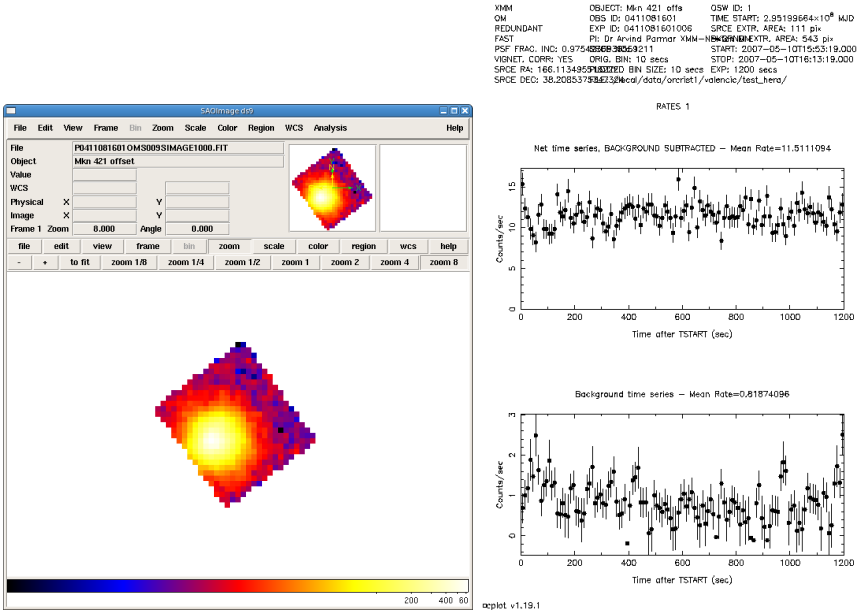
9.4.1 Rerunning the Pipeline

The repipelining task for OM data taken in fast mode is *omfchain*. It produces images of the detected sources, extracts events related to the sources and the background, and extracts the corresponding light curves. At present, unlike *omichain*, *omfchain* does not allow for keywords to specify filters or exposures; calling this task will process all fast mode data.

To run the pipeline on fast mode data with the GUI:

1) In the Available Tools panel, under "XMM-SAS", call *omfchain*.

Figure 9.2: Left: The processed FAST mode sky image. Right: the light curve produced automatically by *omfchain*.



2) In the task pop-up window, click "Run".

To run the pipeline on fast mode data in the Command Window, type:

```
omfchain
```

There are two types of output files: those that start with `f` are intermediate images or time series files; those that start with `p` are products. The output files are are described in Table 9.2.
To demonstrate some of these output files, we have rerun the pipeline on the example dataset and downloaded the output to a local machine. The processed image in sky-coordinates from one exposure, `P0411081601OMS006SIMAGE1000.FIT`, is shown in Fig. 9.2 (left). The background-subtracted light curve produced automatically by the task, `F0411081601OMS006TIMESR1000.PS`, is shown in Fig. 9.2 (right).

9.4.2 Verifying the Output

A good first step in checking the output is to examine the light curve plot for both the source and background, making sure they are reasonable: no isolated, unusually high (or low) values, and no frequent drop-outs. Users should also check the image with *fv* or *ds9* in the Fast mode window to see if the source is near an edge. If it is, it's a good idea to examine the light curves from diffent exposures to verify that they are consistent from exposure to exposure (while keeping in mind any intrinsic source variability). If the image is blurred or unusual in any way, users should check the tracking history file to verify the tracking was reliable.

9.5 Grism Analysis

9.5.1 Rerunning the Pipeline

The repipelining task for OM data taken in grism mode is *omgchain*. It produces images of the detected sources and background, extracts source spectra and region files, and makes source lists and postscript and PDF plots. At present, unlike *omichain*, *omgchain* does not allow for keywords to specify filters or exposures; calling this task will process all grism mode data.

To run the pipeline on grism mode data with the GUI:

1) In the Available Tools panel, under "XMM-SAS", call *omgchain*.

2) In the task pop-up window, click "Run".

To run the pipeline on fast mode data in the Command Window, type:

```
omgchain
```

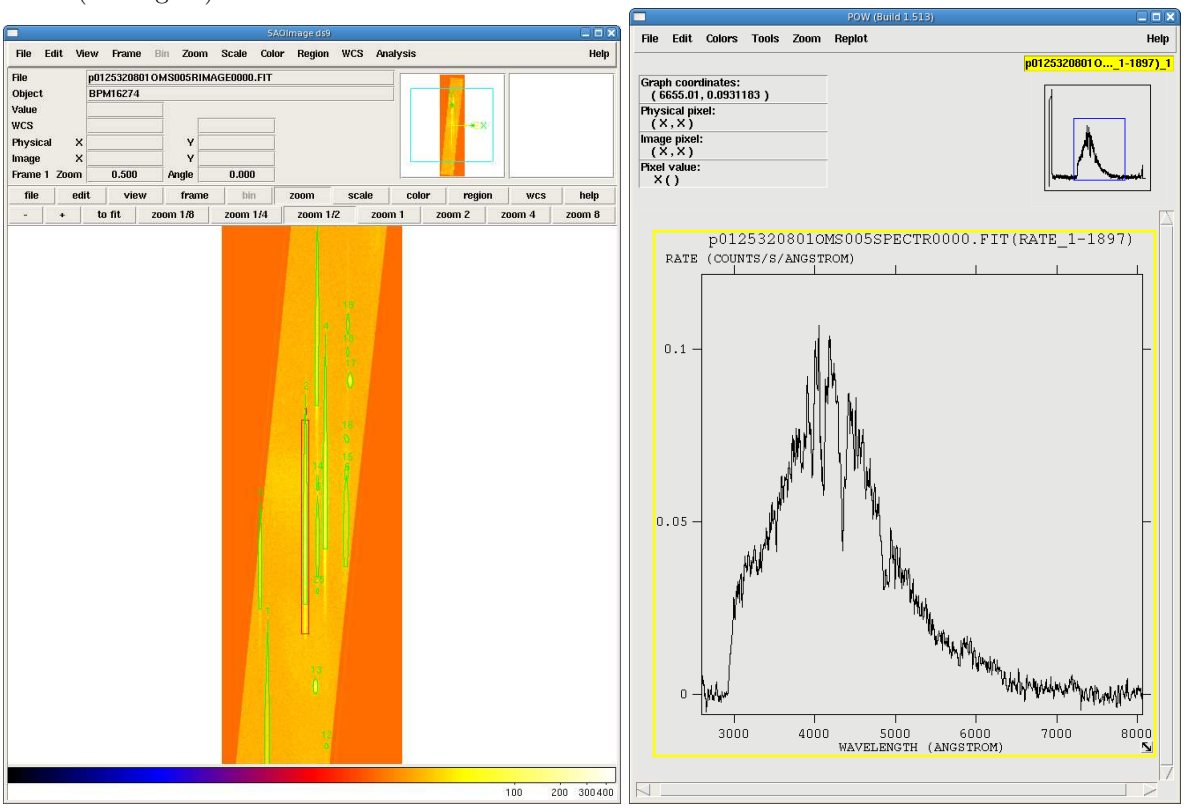
There are two types of output files: those that start with `g` are intermediate or auxiliary files and source lists; those that start with `p` are products. The output files are are described in Table 9.2.

To demonstrate some of these output files, we have rerun the pipeline on the example dataset and downloaded the output to a local machine. The processed image, rotated to align with the columns of the image (`p0125320801OMS005RIMAGE0000.FIT`), is shown in Fig. 9.3 (left). Two region files are overlayed: `p0125320801OMS005REGION0001.ASC`, which corresponds to the sources detected in this rotated image (green), and `p0125320801OMS005SPCREG0001.ASC`, which corresponds to the sources in the spectra list file (red) and indicates the locations of the zero and first orders. The task *omgchain* automatically extracted the spectrum of the red region (`p0125320801OMS005SPECTR0000.FIT`); this is shown in Fig. 9.3 (right).

9.5.2 Verifying the Output

The correct correlation of zero and first orders is crucial for grism analysis. Users should inspect the rotated image with *fv* or *ds9* and verify the identification of the orders by overlaying the `*SPCREG*` region file, as shown in Fig. 9.3 (left); the `*SPECLI*` file also contains this informtaion. If users are interested in all source detections, the region file can also be overlayed and the full source list examined. Users should also examine the spectra plots automatically produced by *omgchain*, for both the source and background, making sure they are reasonable. For improved source detection, the parameter `nsigma` (in the GUI, "Number of sigma for the detection algorithm") can be changed.

Figure 9.3: Left: The repipelined, rotated image with regions overlayed. Right: the spectrum extracted from the source (red region).



9.6 In a Nutshell

To summarize, the following steps are needed to process OM data:

1) Obtain the raw and pipelined data and store them on the Hera server.
2) Initialize Hera.
3) Make the calibration and ODF summary file (run the *cifbuild* and *odfingest* tasks).
4) Rerun the appropriate pipline chain (*omichain*, *omgchain*, or *omfchain*).
5) Examine the output products and verify their quality.